MSX

INCLUINDO BASIC – MSX AVANÇADO

MELHORES PROGRAMAS

EBRAS Editora Brasileira

NELSON ALVES

MSX

MELHORES PROGRAMAS

MSX

MELHORES PROGRAMAS

NELSON ALVES

EBRAS Editora Brasileira

Editor: Ricardo Reinprecht
Produtor Gráfico: Herminio Comis Filho
Área Comercial: José Siqueira Costa
Capa: Ricardo Reinprecht
Arte: Cyro Giordano
Fotocomposição: José Acácio de Campos

Todos os comandos dos programas deste livro foram testados na Staff Consultoria e Processamento Ltda. Entretanto, devido a possibilidade de erros, por menor que seja, acontecer, aceitaremos qualquer sugestão e/ou crítica que venha a melhorar a obra em suas edições posteriores.

1986

EBRAS Editora Brasileira, Importação, Exportação e Comércio de Livros Ltda.
Rua Borges Lagoa, 1.044 — CEP 04038 — Vila Mariana — Fone: (011) 549-8633 —
Telex DRBN (011) 25966 — São Paulo - SP.

Sumário

Sumário

AGRADECIMENTOS

Agradeço à minha esposa, Luciane pelo grande apoio que me deu durante o desenvolvimento deste livro e também ao meu editor Ricardo Reinprecht pela paciência e apoio a mim dispensados.

INTRODUÇÃO

Este livro foi escrito para todos os usuários que desejam utilizar os microcomputadores MSX para aprender mais sobre programação em BASIC.

O livro contém programas práticos e de entretenimento desenvolvidos para auxiliar os usuários no aprendizado do BASIC MSX. Outros livros normalmente contêm exemplos insuficientes ou incompletos, o que dificulta a compreensão da potencialidade do BASIC MSX. A utilização de **MSX — Melhores programas** é, além de fácil, divertida.

Explicações claras e concisas do que está acontecendo em cada programa são fornecidas. À medida que você lê estas explicações, mais você aprende sobre porque o programa trabalha desta maneira e começa a visualizar as técnicas do BASIC MSX que podem ser úteis no desenvolvimento de seus próprios programas.

Com a utilização de um simples microcomputador MSX com um gravador cassete, você poderá executar os programas contidos no **MSX — Melhores Programas** sem nenhum problema. (Mas, se você não quer digitar todos estes programas é possível adquirir uma fita cassete já gravada com todos os programas digitados e testados para a sua tranqüilidae.) Se você tiver uma impressora, muitos programas podem ser utilizados com ela, sem que seja necessária nenhuma mudança nos programas.

O **MSX — Melhores Programas** é um livro único, pois ele tem um grande número de programas que abrangem as mais diversas áreas que o usuário do microcomputador MSX pode ter interesse. Isto lhe dá o apoio para que você possa usar o seu microcomputador MSX fazendo uso de todas as suas características.

Para digitar qualquer programa simples siga as instruções no Manual de Linguagem BASIC. Se você quiser modificar qualquer linha, por exemplo, colocar seus próprios dados na linha de comando DATA, é só fazê-lo. (Certifique-se de fazer a substituição corretamente e de finalizar um comando da mesma maneira que ele está sendo finalizado no programa.)

A ORGANIZAÇÃO DESTE LIVRO

O Capítulo 1 apresenta uma descrição do teclado do MSX, incluindo suas teclas de função, o teclado numérico separado e as teclas de edição. Os Capítulos 2, 3 e 4 contêm diferentes tipos de programas, incluindo problemas matemáticos e sua resolução, jogos e programas práticos e programas educacionais e pessoais. Devido à versatilidade dos programas, não foi fácil fazer a divisão destes em grupos, pois, programas de matemática podem também ser programas educacionais, alguns programas pessoais poderiam ser classificados como programas de economia etc.

Para executar um programa você tem duas maneiras:

1. Digitar o programa assim como ele é mostrado no livro, depois salvá-lo para não precisar redigitá-lo; ou

2. Adquirir a fita cassete opcional que foi desenvolvida e testada para facilitar o uso deste livro. Esta fita já contém todos os programas apresentados neste livro.

Após você decidir qual a maneira que vai utilizar (digitar o programa ou carregá-lo da fita), deve usar o comando **RUN** para executar o programa.

Além dos métodos descritos no livro para encerrar a execução dos programas, você pode finalizar qualquer programa, sem danificá-lo, através do comando **CONTROL+STOP** (ou seja, você pressiona a tecla **CONTROL** e a tecla **STOP** ao mesmo tempo).

Sempre que um programa pedir a você que insira um dado, através de um comando **INPUT**, você deverá seguir esta entrada de dados de um **RETURN**.

Apesar do grande esmero no desenvolvimento e teste na preparação dos programas do **MSX — Melhores Programas,** nem ele nem os programas são necessariamente perfeitos. Mas, cada programa foi testado completamente. Caso o leitor ache qualquer falha, nos colocamos à disposição para receber sua solução ou comentários dela nos endereços abaixo:

EBRAS Editora Brasileira Ltda.
Rua Borges Lagoa, 1.044
04038 — São Paulo - SP
(011) 549 - 8633

ou

STAFF Consultoria e Processamento Ltda.
Rua Pascal, 799
04616 — São Paulo - SP
(011) 530 - 3741

Capítulo 1

O TECLADO DO MSX

INTRODUÇÃO

O teclado é um periférico padrão para a entrada de dados. A maioria das informações que o computador utiliza vêm originalmente do teclado. Usando várias combinações das 89 teclas, você pode digitar qualquer um dos 256 caracteres que o MSX reconhece.

Se você conhece o teclado de uma máquina de escrever comum, as teclas de letras e números não devem causar nenhum problema. Algumas das teclas de pontuação podem estar em diferentes localizações, as quais não são aquelas que você esperava, mas você se adaptará rapidamente à nova configuração.

TECLAS DE CONTROLE GERAL

O teclado do MSX possui algumas teclas que realizam funções de controle geral como descrito abaixo. Os programas aplicativos podem ser desenvolvidos de modo que disabilitem ou mesmo redefinam algumas destas teclas de controle geral; consulte sempre as instruções de operação do programa.

ESC — É uma abreviação para **"ESCape"**. Esta tecla faz o MSX interromper a operação de um programa quando pressionada pela primeira vez, quando for pressionada pela segunda vez ela continua o programa a partir de onde foi interrompido. Ela pode ser utilizada em alguns programas aplicativos.

TAB — A tecla de **TABulação** funciona como o tabulador de uma máquina de escrever comum, movendo o cursor para frente oito espaços até a próxima tabulação.

CONTROL — Às vezes abreviada como **Ctrl**, mais parecida com a tecla de letras maiúsculas de uma máquina de escrever comum, a tecla **CONTROL** modifica o efeito de outra tecla quando as duas são pressionadas ao mesmo tempo, assim como **CONTROL+S-TOP** que interrompe a execução de um programa no BASIC MSX.

SHIFT — Use ambas as teclas **SHIFT** como você usa em uma máquina de escrever comum para acessar as letras maiúsculas ou os símbolos e sinais de pontuação. Quando o teclado está ativado no modo de letra maiúscula (através da tecla **CAPS LOCK**), a tecla **SHIFT** age como se não estivesse acionada, retornando letras maiúsculas quando você a mantém pressionada; elas não desativam o modo de letras maiúsculas.

BS — É a abreviatura de **BackSpace**.Funciona como o retorno de uma letra de uma máquina de escrever comum. Embora, ela não mova somente o cursor um caracter para trás, mas também apaga o caracter que estava nesta posição.

RETURN — O mais semelhante à esta telcla é o "retorno do carro" encontrado em algumas máquinas de escrever elétricas. Em alguns computadores, a tecla ↵ é chamada de **RETURN** ou **ENTER**. Seu objetivo é finalizar a linha que você está digitando e avançar para a próxima linha.

CAPS LOCK — A tecla **CAPS LOCK** faz com que o teclado gere letras maiúsculas sem a necessidade de se utilizar a tecla **SHIFT**. De fato, quando o teclado está no modo de letras maiúsculas, utilizando-se a tecla **SHIFT** ele retornará letras minúsculas. A tecla **CAPS LOCK** difere da tecla correspondente da máquina de escrever porque ela afeta somente as 26 teclas que possuem as letras do alfabeto. Por exemplo, você deve sempre pressionar ambas **SHIFT** e / para ter o símbolo de interrogação (?). Pressionando-se uma vez a tecla **CAPS LOCK** muda-se para o modo de letra maiúscula; pressionando-a novamente retorna-se ao modo de letra minúscula. A tecla **C** não é ativada, para obtê-la deve-se pressionar a tecla **SHIFT** juntamente com C.

HOME/CLS — O canto esquerdo superior da tela é chamado de posição (inicial). **("home"). A tecla HOME/CLS** move o cursor para lá.Se ela for pressionada junto com a tecla **SHIFT** a função **CLS** é ativada e então, além de mover o cursor para o canto superior esquerdo da tela ela a limpa (ou seja, "zera" a tela).

LGRA — Quando você quer gerar os símbolos gráficos diretamente sem precisar procurar qual é o código ASCII equivalente, pode utilizar esta tecla, mais a que gera o símbolo solicitado. (Consulte o Manual do seu microcomputador para saber quais são as teclas).

RGRA — Tem o mesmo efeito da tecla **LGRA** apenas gerando caracteres diferentes.

INSERT — A tecla **INSERT** coloca e tira o computador do modo de inserção. No modo de inserção, o cursor é um traço ao invés de um pequeno quadrado, e cada caracter que você digitar "empurrará" os restantes dos caracteres uma posição para a direita. Isto é melhor do que ter que reescrever uma linha inteira.

DELETE — A tecla **DELETE** apaga o caracter sobre o qual o cursor está. Todos os caracteres à direita de um caracter apagado são então movidos uma posição para a esquerda para preencher o espaço que foi deixado vago quando se apagou o caracter.

STOP — Esta tecla junto com a tecla **SHIFT** tem a função parecida com a de **CTRL+S** de alguns computadores, ou seja, ela interrompe e continua a execução de um programa ou a visualização de

uma listagem de programa. Para interromper definitivamente a execução de um programa deve-se utilizar a tecla **STOP** junto com a tecla **CONTROL**. Isto retornará o comando ao usuário.

SELECT Esta tecla não tem função quando o usuário está programando em BASIC MSX. Ela serve para quando se está utilizando programas de processamento de palavras ou na entrada de dados quando se tem programas aplicativos que a utilizam.

SETAS Estas teclas servem para mover o cursor para qualquer parte da tela. Também servem para edição de programas em BASIC MSX, facilitando assim a correção de erros em linhas do programa.

F1...F10 As teclas de função **F1...F10** na parte superior do teclado, podem ser programadas para gerar seqüências de caracteres diferentes. Alguns programas aplicativos podem usar as teclas de função para um único propósito; consultar suas instruções de operação para detalhes específicos. Quando você estiver utilizando o BASIC MSX estas teclas já estarão pré-definidas (você pode mudá-las se achar conveniente).

F1	**COLOR**
F2	**AUTO**
F3	**GOTO**
F4	**LIST**
F5	**RUN** (não precisa de **RETURN**)
F6	**COLOR** 15,1,1
F7	**CLOAD**
F8	**CONT**
F9	**LIST**
F10	**RUN** (precisa de **RETURN**)

REPETIÇÃO AUTOMÁTICA DAS TECLAS

Você deve ter descoberto que se pressionar a tecla e mantê-la pressionada, ela será repetida automaticamente. Esta característica funciona com todas as teclas exceto **CONTROL**, **SHIFT** e **INSERT** quando pressionadas sozinhas. Embora combinações destas teclas com outras sejam repetidas.

DIGITANDO COMANDOS

Quando aparentemente não existe nenhum sistema ativo por trás do cursor, o MSX está provavelmente aguardando um comando. Você executa comandos digitando uma seqüência de teclas. O computador, dirigido pelo programa, examina o que você digitou e tenta determinar o que fazer. Digite algo corretamente e o programa fará com que o computador carregue o seu comando. Assim, é o programa e não o próprio computador, que determina a validade dos

comandos que você digitou.

Cada programa tem o seu próprio conjunto de comandos, e quando ele controla o MSX, estes são os únicos comandos que o MSX entende. Pode ser um programa aplicativo que está sendo controlado — por exemplo, um processador de texto. Por outro lado, o interpretador BASIC MSX pode estar no controle. Isto quer dizer que precisamos saber qual o programa que está controlando o MSX quando nos preparamos para digitar um comando.

Um comando pode não ser exclusivo de um único pograma. Mas, tenha cuidado, porque mesmo que um comando pareça o mesmo, seu efeito pode diferir de acordo com a tarefa que realiza. Em outras palavras, uma certa seqüência de teclas pode significar uma coisa em um programa, a mesma coisa para outro programa, algo inteiramente diferente para um terceiro programa, e nada para um quarto programa.

Devido ao tamanho dos comandos ser variável, não existe um limite prático para seu tamanho. O programa que está no controle não age até que terminemos de digitar. O caracter que avisa que terminamos de digitar um comando geralmente é o **↵ (RETURN)**. Portanto, se você digitar um comando e ficar sentado esperando que algo aconteça, provavelmente esqueceu de digitar **RETURN**. Existem comandos que não precisam de finalizadores e muitos poucos que utilizam outros tipos de finalizadores, por isto, excessões como estas devem estar bem documentadas nos manuais dos programas que os utilizam.

CORRIGINDO DIGITAÇÕES ERRADAS

Cometer um erro e o computador fazer algo errado ou mesmo nada, acontece com todo mundo. Poucas pessoas nunca cometem erros, por isto o MSX fornece meios para corrigir erros. Para erros que você notou antes de pressionar **RETURN**, use a tecla **BackSpace (BS)** para corrigí-lo. quando notar que o erro está no início de um comando ou de uma linha de comando, faça uso das teclas com as setas que estão localizadas do lado direito inferior de seu teclado. Posicione o cursor embaixo do caracter errado e digite o caracter correto.

ACESSANDO OS CARACTERES GRÁFICOS DIRETAMENTE

Se você desejar fazer desenhos na tela, ou colocar algum caracter gráfico em seu programa aplicativo, deve usar as teclas **LGRA** e **RGRA**. Estas teclas sozinhas e também junto com a tecla **SHIFT** permitem que você acesse todos os caracteres contidos na tabela ASCII que está na EPROM do seu computador.

Para saber quais caracteres são acessados veja o manual do seu computador.

Capítulo 2

JOGOS E PROGRAMAS PRÁTICOS

GERAÇÃO DE CARACTER

O programa **Geração de Caracter (CARACS)** permite a você mostrar um caracter (ou um sinal), em tamanho maior, no sentido inverso da tela e gerar uma saida do mesmo para a sua impressora.

O comando **DIM** nas linhas 120 - 140 define as três matrizes de caracteres. A$ contém os 26 caracteres alfabéticos (eles devem ser digitados com a tecla **CAPS LOCK** acionada), N$ contém os 10 dígitos numéricos de 0 a 9 e P$ contém os valores dos caracteres de pontuação (ponto-e-vírgula, dois-pontos, ponto, hífen, exclamação, asterisco, ponto-de-interrogação e o sinal de mais).

As linhas 200 - 270 lêem **(READ)** os dados que mostram os caracteres alfabéticos. Como cada linha é lida de um comando **DATA,** ele é colocado em uma matriz bidimensional A$. Isto é, a primeira linha de caracteres, por exemplo, A's vai em A$(1,1), a seguir em A$(1,2) etc., até o valor **"CARACTER"** ser lido; para o próximo caracter, a primeira linha de B's é colocada em A$(2,1), em A$(2,2); e assim por diante, até o valor "caracter" ser lido. Após cada caracter ser lido, C1, a primeira variável subscrita, é incrementada de 1; e C2, a segunda variável subscrita, é colocada em 0. Assim, a letra A está contida em A$(1,1) até A$(1,9), e assim por diante, até a letra Z, a qual está nas variáveis A$(26,1) até A$(26,11). Quando a linha 230 é lida **(READ)** o "FIM" do comando **DATA** da linha 3.990 indica que a carga dos caracteres alfabéticos foi realizada e o controle passa para a linha 280 para começar a carga dos dígitos numéricos.

O dígito numérico é lido **(READ)**, assim como os caracteres de pontuação nas linhas 280 - 350 e 360 - 430, são carregados de modo similar; mas, naturalmente, as matrizes N$ e P$ têm os valores apropriados. Se você respondeu S à questão da linha 160, o comando **DATA** pode ser mostrado na tela à medida que ele vai carregando a memória.

As linhas 440 - 500 então permitem que você digite uma mensagem para ser mostrada na tela ou impressa em sua impessora em letras maiores. Digitando FIM, você pode parar o programa.

Nas linhas 580 a 1.030, a análise caracter-por-caracter da mensagem é feita, fazendo com que a linha apropriada para aumentar o caracter seja chamada da memória. As linhas 580 a 600 iniciam a análise. O programa procura por uma mensagem inteira, da esquerda para a direita. A função **MID$** isola cada caracter; por exemplo MID$(B$,2,1) procura o segundo caracter de B$ (a mensagem que foi digitada) para o comprimento de 1 caracter. Então o valor ASCII correspondente àquele caracter é colocado na variável A. Se o valor de X$ está entre A e Z, a linha 620 pula para a rotina de caracteres alfabéticos que começa na linha 790. A linha 640 envia os valores numéricos para a linha 870, e todos os outros pulam para a linha 660 para avaliação da pontuação.

Note que 64 ou 47 é subtraido do valor ASCII (para caracteres alfabéticos e valores numéricos, respectivamente). O resultado deste cálculo é que A agora

contém a própria variável subscrita (de 1 a 26 oude 1 a 10) e pode carregar a informação do caracter aumentado).

Os comandos nas linhas 660 - 730 dão aos caracteres de pontuação o valor subscrito apropriado. Qualquer outro caracter é inválido e causa o surgimento de um espaço (10 linhas em branco, como definido nas linhas 740 - 780). As linhas 790 - 860 imprimem **(PRINT/LPRINT)** os caracteres A a Z; as linhas 870 - 940 imprimem **(PRINT / LPRINT)** os números 0 a 9; e as linhas 950 - 1.020 imprimem **(PRINT / LPRINT)** os caracteres de pontuação. A linha 1.030 a seguir transfere o controle de novo para a linha 440 para criar outra mensagem ou finalizar o programa.

Observando os comandos **DATA** das linhas 1.040 - 5.860, você pode facilmente ver como o caracter foi criado. A altura e a aparência de cada caracter pode ser modificada se desejado.

```
10 REM  ***********************
20 REM  *                     *
30 REM  * GERACAO DE CARACTER *
40 REM  *                     *
50 REM  *  DIREITOS AUTORAIS  *
60 REM  *                     *
70 REM  *     NELSON ALVES    *
80 REM  *                     *
90 REM  *                     *
100 REM ***********************
110 KEY OFF
120 DIM A$(30,16)
130 DIM N$(15,16)
140 DIM P$(15,16)
150 CLS
160 INPUT "LISTAR NA TELA COMO MATRIZ (S/N) ";AY$
170 PRINT: PRINT "CARREGANDO AS MATRIZES..."
180 C1 = 1
190 C2 = 0
200 READ X$
210 IF AY$ <> "S" THEN 230
220 IF X$ <> "CARACTER" AND X$ <> "FIM" THEN PRINT X$
230 IF X$ = "FIM" THEN C1 = 1: C2 = 0: GOTO 280
240 C2 = C2+1
250 A$(C1,C2) = X$
260 IF X$ = "CARACTER" THEN C1 = C1+1: C2 = 0
270 GOTO 200
280 READ X$
```



```
290 IF AY$ <> "S" THEN 310
300 IF X$ <> "CARACTER" AND X$ <> "FIM" THEN PRINT X$
310 IF X$ = "FIM" THEN C1 = 1: C2 = 0: GOTO 360
320 C2 = C2 + 1
330 N$(C1,C2) = X$
340 IF X$ = "CARACTER" THEN C1 = C1+1: C2 = 0
350 GOTO 280
360 READ X$
370 IF AY$ <> "S" THEN 390
380 IF X$ = "CARACTER" AND X$ = "FIM" THEN PRINT X$
390 IF X$ = "FIM" THEN C1 = 1: C2 = 0: GOTO 440
400 C2 = C2 + 1
410 P$(C1,C2) = X$
420 IF X$ = "CARACTER" THEN C1 = C1 + 1: C2 = 0
430 GOTO 360
440 CLS
450 LOCATE 0,6
460 INPUT "IMPRIMIR (S/N) ";S$
470 PRINT
480 PRINT "(FIM PARA TERMINAR O PROGRAMA)"
490 PRINT
500 INPUT "QUAL O CARACTER DESEJADO ";B$
510 IF B$ = "FIM" THEN 5870
520 IF S$ <> "S" THEN 580
530 PRINT
540 INPUT "IMPRESSAO (N)ORMAL OU (C)ONDENSADA ";RY$
550 LPRINT CHR$(27);CHR$(78);
560 IF RY$ = "N" THEN 580
570 LPRINT CHR$(27);CHR$(81);
580 FOR I = 1 TO LEN(B$)
590 X$ = MID$(B$,1,1)
600 A = ASC(X$)
610 REM *** LETRAS ***
620 IF X$ >= "A" AND X$ <= "Z" THEN A= A-64: GOTO 790
630 REM *** NUMEROS ***
640 IF X$ >= "0" AND X$ <= "9" THEN A = A-47: GOTO 870
650 REM *** PONTUACAO ***
660 IF X$ = ";" THEN A = 2: GOTO 950
670 IF X$ = ":" THEN A = 3: GOTO 950
680 IF X$ = "." THEN A = 4: GOTO 950
690 IF X$ = "-" THEN A = 5: GOTO 950
700 IF X$ = "!" THEN A = 6: GOTO 950
710 IF X$ = "*" THEN A = 7: GOTO 950
720 IF X$ = "?" THEN A = 8: GOTO 950
730 IF X$ = "+" THEN A = 9: GOTO 950
```

```
740 REM ** ESPACO ***
750 FOR J = 1 TO 10
760 LPRINT
770 NEXT J
780 GOTO 1020
790 FOR J = 1 TO 15
800 IF A$(A,J) = "CARACTER" THEN 860
810 IF S$ <> "S" THEN 840
820 LPRINT SPC(10);A$(A,J)
830 GOTO 850
840 PRINT SPC(10);A$(A,J)
850 NEXT J
860 GOTO 1020
870 FOR J = 1 TO 15
880 IF N$(A,J) = "CARACTER" THEN 860
890 IF S$ <> "S" THEN 920
900 LPRINT SPC(10);N$(A,J)
910 GOTO 930
920 PRINT SPC(10);N$(A,J)
930 NEXT J
940 GOTO 1020
950 FOR J = 1 TO 15
960 IF P$(A,J) = "CARACTER" THEN 860
970 IF S$ <> "S" THEN 1000
980 LPRINT SPC(10);P$(A,J)
990 GOTO 1010
1000 PRINT SPC(10);P$(A,J)
1010 NEXT J
1020 NEXT I
1030 FOR I = 1 TO 1000: NEXT I: GOTO 440
1040 DATA ""
1050 DATA "AAAAAAAAAAAAAAA"
1060 DATA "AAAAAAAAAAAAAAAA"
1070 DATA "        AA    AA"
1080 DATA "        AA    AA"
1090 DATA "        AA    AA"
1100 DATA "AAAAAAAAAAAAAAAA"
1110 DATA "AAAAAAAAAAAAAAA"
1120 DATA ""
1130 DATA "CARACTER"
1140 DATA ""
1150 DATA "BBBBBBBBBBBBBBBB"
1160 DATA "BBBBBBBBBBBBBBBB"
1170 DATA "BB     BB     BB"
1180 DATA "BB     BB     BB"
```



```
1190 DATA " BB   BBBB    BB"
1200 DATA "  BBBBB  BBBBB"
1210 DATA "   BBB    BBB"
1220 DATA ""
1230 DATA "CARACTER"
1240 DATA ""
1250 DATA " CCCCCCCCCCCCCC"
1260 DATA "CCCCCCCCCCCCCCCC"
1270 DATA "CCC          CCC"
1280 DATA "CC            CC"
1290 DATA "CC            CC"
1300 DATA "CC            CC"
1310 DATA "CCC          CCC"
1320 DATA ""
1330 DATA "CARACTER"
1340 DATA ""
1350 DATA "DDDDDDDDDDDDDDDD"
1360 DATA "DDDDDDDDDDDDDDDD"
1370 DATA "DD            DD"
1380 DATA "DDD          DDD"
1390 DATA " DDD        DDD"
1400 DATA "  DDD      DDD"
1410 DATA "   DDDD  DDDD"
1420 DATA "    DDDDDDDD"
1430 DATA "      DDDD"
1440 DATA ""
1450 DATA "CARACTER"
1460 DATA ""
1470 DATA "EEEEEEEEEEEEEEEE"
1480 DATA "EEEEEEEEEEEEEEEE"
1490 DATA "EE    EE     EE"
1500 DATA "EE    EE     EE"
1510 DATA "EE    EE     EE"
1520 DATA "EE    EE     EE"
1530 DATA "EE    EE     EE"
1540 DATA ""
1550 DATA "CARACTER"
1560 DATA ""
1570 DATA "FFFFFFFFFFFFFFFF"
1580 DATA "FFFFFFFFFFFFFFFF"
1590 DATA "      FF     FF"
1600 DATA "      FF     FF"
1610 DATA "      FF     FF"
1620 DATA "      FF     FF"
1630 DATA "      FF     FF"
```

```
1640 DATA ""
1650 DATA "CARACTER"
1660 DATA ""
1670 DATA "GGGGGGGGGGGGGGGG"
1680 DATA "GGGGGGGGGGGGGGGG"
1690 DATA "GGG          GGG"
1700 DATA "GG            GG"
1710 DATA "GG            GG"
1720 DATA "GG  GG        GG"
1730 DATA "GGGGGG        GG"
1740 DATA "GGGGGG        GG"
1750 DATA ""
1760 DATA "CARACTER"
1770 DATA ""
1780 DATA "HHHHHHHHHHHHHHHH"
1790 DATA "HHHHHHHHHHHHHHHH"
1800 DATA "       HH"
1810 DATA "       HH"
1820 DATA "       HH"
1830 DATA "       HH"
1840 DATA "HHHHHHHHHHHHHHHH"
1850 DATA "HHHHHHHHHHHHHHHH"
1860 DATA ""
1870 DATA "CARACTER"
1880 DATA ""
1890 DATA "II            II"
1900 DATA "II            II"
1910 DATA "IIIIIIIIIIIIIIII"
1920 DATA "IIIIIIIIIIIIIIII"
1930 DATA "II            II"
1940 DATA "II            II"
1950 DATA ""
1960 DATA "CARACTER"
1970 DATA ""
1980 DATA "JJJJ"
1990 DATA "JJJJ"
2000 DATA "JJJ"
2010 DATA "JJ"
2020 DATA "JJ"
2030 DATA "JJJJJJJJJJJJJJJJ"
2040 DATA "JJJJJJJJJJJJJJJJ"
2050 DATA ""
2060 DATA "CARACTER"
2070 DATA ""
2080 DATA "KKKKKKKKKKKKKKKK"
```



```
2090 DATA "KKKKKKKKKKKKKKKK"
2100 DATA "      KKKK"
2110 DATA "     KK  KK"
2120 DATA "    KK    KK"
2130 DATA "   KK      KK"
2140 DATA "  KK        KK"
2150 DATA " KK          KK"
2160 DATA "KK            KK"
2170 DATA ""
2180 DATA "CARACTER"
2190 DATA ""
2200 DATA "LLLLLLLLLLLLLLLL"
2210 DATA "LLLLLLLLLLLLLLLL"
2220 DATA "LLL"
2230 DATA "LL"
2240 DATA "LL"
2250 DATA "LL"
2260 DATA "LL"
2270 DATA "LL"
2280 DATA ""
2290 DATA "CARACTER"
2300 DATA ""
2310 DATA "MMMMMMMMMMMMMMMM"
2320 DATA "MMMMMMMMMMMMMMMM"
2330 DATA "            MM"
2340 DATA "           MM"
2350 DATA "          MM"
2360 DATA "         MM"
2370 DATA "         MM"
2380 DATA "          MM"
2390 DATA "           MM"
2400 DATA "            MM"
2410 DATA "MMMMMMMMMMMMMMMM"
2420 DATA "MMMMMMMMMMMMMMMM"
2430 DATA ""
2440 DATA "CARACTER"
2450 DATA ""
2460 DATA "NNNNNNNNNNNNNNNN"
2470 DATA "NNNNNNNNNNNNNNNN"
2480 DATA "           NNN"
2490 DATA "         NNN"
2500 DATA "       NNN"
2510 DATA "     NNN"
2520 DATA "   NNN"
2530 DATA " NNN"
```

```
2540 DATA "NNNNNNNNNNNNNNNN"
2550 DATA "NNNNNNNNNNNNNNNN"
2560 DATA ""
2570 DATA "CARACTER"
2580 DATA ""
2590 DATA " OOOOOOOOOOOOOO "
2600 DATA "OOOOOOOOOOOOOOOO"
2610 DATA "OO            OO"
2620 DATA "OO            OO"
2630 DATA "OO            OO"
2640 DATA "OO            OO"
2650 DATA "OO            OO"
2660 DATA "OOOOOOOOOOOOOOOO"
2670 DATA " OOOOOOOOOOOOOO "
2680 DATA ""
2690 DATA "CARACTER"
2700 DATA ""
2710 DATA "PPPPPPPPPPPPPPPP"
2720 DATA "PPPPPPPPPPPPPPPP"
2730 DATA "        PP    PP"
2740 DATA "        PP    PP"
2750 DATA "        PP    PP"
2760 DATA "        PP    PP"
2770 DATA "        PPPPPPPP"
2780 DATA "         PPPPPP"
2790 DATA ""
2800 DATA "CARACTER"
2810 DATA ""
2820 DATA " QQQQQQQQQQQQQQ "
2830 DATA "QQQQQQQQQQQQQQQQ"
2840 DATA "QQ            QQ"
2850 DATA "QQ            QQ"
2860 DATA "QQ            QQ"
2870 DATA "QQ QQ         QQ"
2880 DATA "QQQQ          QQ"
2890 DATA "QQQQQQQQQQQQQQQQ"
2900 DATA "QQQQQQQQQQQQQQQ"
2910 DATA "QQ"
2920 DATA "QQ"
2930 DATA ""
2940 DATA "CARACTER"
2950 DATA ""
2960 DATA "RRRRRRRRRRRRRRRR"
2970 DATA "RRRRRRRRRRRRRRRR"
2980 DATA "       RR     RR"
```



```
2990 DATA "       RR     RR"
3000 DATA "     RRRR     RR"
3010 DATA "   RRR RR     RR"
3020 DATA " RRR   RRRRRRRRR"
3030 DATA "RR      RRRRRRR"
3040 DATA ""
3050 DATA "CARACTER"
3060 DATA ""
3070 DATA "SS      SSSSSSS "
3080 DATA "SS     SSSSSSSSS"
3090 DATA "SS     SS     SS"
3100 DATA "SS     SS     SS"
3110 DATA "SS     SS     SS"
3120 DATA "SSSSSSSSS     SS"
3130 DATA " SSSSSSS      SS"
3140 DATA ""
3150 DATA "CARACTER"
3160 DATA ""
3170 DATA "              TT"
3180 DATA "              TT"
3190 DATA "              TT"
3200 DATA "TTTTTTTTTTTTTTTT"
3210 DATA "TTTTTTTTTTTTTTTT"
3220 DATA "              TT"
3230 DATA "              TT"
3240 DATA "              TT"
3250 DATA ""
3260 DATA "CARACTER"
3270 DATA ""
3280 DATA " UUUUUUUUUUUUUUU"
3290 DATA "UUUUUUUUUUUUUUUU"
3300 DATA "UU"
3310 DATA "UU"
3320 DATA "UU"
3330 DATA "UU"
3340 DATA "UU"
3350 DATA "UUUUUUUUUUUUUUUU"
3360 DATA " UUUUUUUUUUUUUUU"
3370 DATA ""
3380 DATA "CARACTER"
3390 DATA ""
3400 DATA "            VVVV"
3410 DATA "         VVVV"
3420 DATA "      VVVV"
3430 DATA "   VVVV"
```

```
3440 DATA "VVVV"
3450 DATA "VV"
3460 DATA "VVVV"
3470 DATA "   VVVV"
3480 DATA "      VVVV"
3490 DATA "         VVVV"
3500 DATA "            VVVV"
3510 DATA ""
3520 DATA ""
3530 DATA "CARACTER"
3540 DATA ""
3550 DATA "WWWWWWWWWWWWWWWW"
3560 DATA " WWWWWWWWWWWWWWW"
3570 DATA "   WWW"
3580 DATA "     WWW"
3590 DATA "       WWW"
3600 DATA "     WWW"
3610 DATA "   WWW"
3620 DATA " WWWWWWWWWWWWWWW"
3630 DATA "WWWWWWWWWWWWWWWW"
3640 DATA ""
3650 DATA "CARACTER"
3660 DATA ""
3670 DATA "XXX          XXX"
3680 DATA "  XXX      XXX"
3690 DATA "    XXX  XXX"
3700 DATA "      XXXX"
3710 DATA "      XXXX"
3720 DATA "    XXX  XXX"
3730 DATA "  XXX      XXX"
3740 DATA "XXX          XXX"
3750 DATA ""
3760 DATA "CARACTER"
3770 DATA ""
3780 DATA "             YYY"
3790 DATA "           YYY"
3800 DATA "         YYY"
3810 DATA "YYYYYYYYYY"
3820 DATA "YYYYYYYYYY"
3830 DATA "         YYY"
3840 DATA "           YYY"
3850 DATA "             YYY"
3860 DATA ""
3870 DATA "CARACTER"
3880 DATA ""
```



```
3890 DATA "ZZ            ZZ"
3900 DATA "ZZZ           ZZ"
3910 DATA "ZZZZZ         ZZ"
3920 DATA "ZZ ZZZ        ZZ"
3930 DATA "ZZ    ZZZ     ZZ"
3940 DATA "ZZ      ZZZ   ZZ"
3950 DATA "ZZ        ZZZ ZZ"
3960 DATA "ZZ          ZZZZ"
3970 DATA "ZZ            ZZ"
3980 DATA ""
3990 DATA "FIM"
4000 DATA ""
4010 DATA " 00000000000000"
4020 DATA "0000000000000000"
4030 DATA "000000        00"
4040 DATA "00    0000    00"
4050 DATA "00        000000"
4060 DATA "0000000000000000"
4070 DATA " 00000000000000 "
4080 DATA ""
4090 DATA "CARACTER"
4100 DATA ""
4110 DATA "11            1"
4120 DATA "11           11"
4130 DATA "1111111111111111"
4140 DATA "1111111111111111"
4150 DATA "11"
4160 DATA "11"
4170 DATA ""
4180 DATA "CARACTER"
4190 DATA ""
4200 DATA "22            22"
4210 DATA "2222          22"
4220 DATA "22 222        22"
4230 DATA "22   222      22"
4240 DATA "22     222    22"
4250 DATA "22       22  222"
4260 DATA "22        22222"
4270 DATA "22         222"
4280 DATA ""
4290 DATA "CARACTER"
4300 DATA ""
4310 DATA "33            33"
4320 DATA "33            33"
4330 DATA "33            33"
```

```
4340 DATA "33     33     33"
4350 DATA "33     33     33"
4360 DATA "333   3333   333"
4370 DATA " 333 333333 333"
4380 DATA "  33333  33333"
4390 DATA ""
4400 DATA "CARACTER"
4410 DATA ""
4420 DATA "    444"
4430 DATA "    4444"
4440 DATA "    44 444"
4450 DATA "    44   444"
4460 DATA "    44     444"
4470 DATA "    44      444"
4480 DATA "444444444444444"
4490 DATA "444444444444444"
4500 DATA "    44"
4510 DATA "    44"
4520 DATA ""
4530 DATA "CARACTER"
4540 DATA ""
4550 DATA "55     55555555"
4560 DATA "55     55555555"
4570 DATA "55     55    55"
4580 DATA "55     55    55"
4590 DATA "555  5555    55"
4600 DATA " 5555555     55"
4610 DATA "  55555      55"
4620 DATA ""
4630 DATA "CARACTER"
4640 DATA ""
4650 DATA " 66666666666666"
4660 DATA "666666666666666"
4670 DATA "66     66    66"
4680 DATA "66     66    66
4690 DATA "66     66    66"
4700 DATA "666666666    66"
4710 DATA " 6666666     66"
4720 DATA "  66666      6"
4730 DATA ""
4740 DATA "CARACTER"
4750 DATA ""
4760 DATA "777          77"
4770 DATA "  777        77"
4780 DATA "    777      77"
```



```
4790 DATA "      777    77"
4800 DATA "        777  77"
4810 DATA "          77777"
4820 DATA "            777"
4830 DATA ""
4840 DATA "CARACTER"
4850 DATA ""
4860 DATA " 888888  888888"
4870 DATA "8888888888888888"
4880 DATA "88     88     88"
4890 DATA "88     88     88"
4900 DATA "88     88     88"
4910 DATA "88     88     88"
4920 DATA "8888888888888888"
4930 DATA " 888888  888888"
4940 DATA ""
4950 DATA "CARACTER"
4960 DATA ""
4970 DATA "99       999999"
4980 DATA "99      99999999"
4990 DATA "99      999  999"
5000 DATA "99      99    99"
5010 DATA "99      99    99"
5020 DATA "999    9999  999"
5030 DATA "9999999999999999"
5040 DATA " 999999999999999"
5050 DATA ""
5060 DATA "FIM"
5070 DATA ""
5080 DATA ""
5090 DATA ""
5100 DATA "CARACTER"
5110 DATA ""
5120 DATA ""
5130 DATA "X"
5140 DATA " XX"
5150 DATA "  XXX   XXX"
5160 DATA "  XXXX  XXX"
5170 DATA "   XXX  XXX"
5180 DATA ""
5190 DATA ""
5200 DATA "CARACTER"
5210 DATA ""
5220 DATA ""
5230 DATA "  XXX   XXX"
```

```
5240 DATA " XXX   XXX"
5250 DATA " XXX   XXX"
5260 DATA ""
5270 DATA ""
5280 DATA "CARACTER"
5290 DATA ""
5300 DATA ""
5310 DATA "XXX"
5320 DATA "XXX"
5330 DATA "XXX"
5340 DATA ""
5350 DATA ""
5360 DATA "CARACTER"
5370 DATA ""
5380 DATA ""
5390 DATA "    XX   "
5400 DATA "    XX   "
5410 DATA "    XX   "
5420 DATA "    XX   "
5430 DATA "    XX   "
5440 DATA "    XX   "
5450 DATA ""
5460 DATA "CARACTER"
5470 DATA ""
5480 DATA ""
5490 DATA "XXX  XXXXXXXX"
5500 DATA "XXX XXXXXXXXXX"
5510 DATA "XXX  XXXXXXXX"
5520 DATA ""
5530 DATA ""
5540 DATA "CARACTER"
5550 DATA ""
5560 DATA "  X  X  X"
5570 DATA "   X  X  X"
5580 DATA "    X X X"
5590 DATA "     XXX"
5600 DATA " XXXXXXXXXXX"
5610 DATA "     XXX"
5620 DATA "    X X X"
5630 DATA "   X  X  X"
5640 DATA "  X  X  X"
5650 DATA ""
5660 DATA "CARACTER"
5670 DATA ""
5680 DATA "XX XXXX    XX"
```



```
5690 DATA "XX XXXX    XX"
5700 DATA "     XX    XX"
5710 DATA "     XX    XX"
5720 DATA "     XXXXXXXX"
5730 DATA "      XXXXXX"
5740 DATA ""
5750 DATA "CARACTER"
5760 DATA ""
5770 DATA "     XX"
5780 DATA "     XX"
5790 DATA "     XX"
5800 DATA "  XXXXXXXX"
5810 DATA "  XXXXXXXX"
5820 DATA "     XX"
5830 DATA "     XX"
5840 DATA "     XX"
5850 DATA "CARACTER"
5860 DATA "FIM"
5870 CLS: END
```

```
FF     FF
FF     FF
FF     FF
FF     FF
FF     FF
FFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFF

FF     FF
FF     FF
FF     FF
FF     FF
FF     FF
FFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFF

 UUUUUUUUUUUUUUU
UUUUUUUUUUUUUUUU
UU    UU
UU    UU
UU    UU
UUUUUUUUUUUUUUUU
 UUUUUUUUUUUUUUU

LL
LL
LL
LLLLLLLLLLLLLLLL
LLLLLLLLLLLLLLLL
LL
LL
LL

SS      SSSSSSS
SS     SSSSSSSSS
SS     SS     SS
SS     SS     SS
SS     SS     SS
SSSSSSSSS     SS
 SSSSSSS      SS
```

Figura 1. Programa Geração de Caracter: exemplo (1) de saída do programa.


```
SS    SSSSSSS
SS   SSSSSSSSS
SS   SS     SS
SS   SS     SS
SS   SS     SS
SSSSSSSSS   SS
 SSSSSSS    SS

UUUUUUUUUUUUUUUU
UUUUUUUUUUUUUUUU
UU    UU
UU    UU
UU    UU
UUUUUUUUUUUUUUUU
 UUUUUUUUUUUUUUU

 RRRRRR     RR
RRRRRRR    RRR
RR     RR RRR
RR     RRRR
RR     RR
RR     RR
RRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRR

   BBB    BBB
  BBBBB  BBBBB
 BB   BBBB   BB
BB     BB     BB
BB     BB     BB
BBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBB

EE     EE     EE
EE     EE     EE
EE     EE     EE
EE     EE     EE
EE     EE     EE
EEEEEEEEEEEEEEEE
EEEEEEEEEEEEEEEE
```

Figura 2. Programa Geração de Caracter: exemplo (2) de saída do programa.

GERAÇÃO DE SOM

O programa **Geração de Som (SOM)** é um programa de entretenimento que permite que você digite qualquer mensagem que deseja ver repetida várias vezes na tela. Ao mesmo tempo em que a mensagem é mostrada, um *beep* ou tom é emitido.

Após você ter selecionadado a mensagem na linha 130, a linha 150 solicita número de *beeps*. Se você quiser que a mensagem seja mostrada por um longo período, digite um número alto. A tela é "limpa" após cada *beep* se você responder S à pergunta da linha 180. Se você quiser repetir a mensagem, digite MESMA para a pergunta "QUAL A MENSAGEM?"; você não será solicitado a informar a quantidade de *beeps* ou a zerar a tela.

```
10 REM  ************************
20 REM  *                      *
30 REM  *    GERACAO DE SOM    *
40 REM  *                      *
50 REM  * DIREITOS AUTORAIS    *
60 REM  *                      *
70 REM  *    NELSON ALVES      *
80 REM  *                      *
90 REM  *                      *
100 REM ************************
110 KEY OFF
120 CLS
130 INPUT "QUAL E A MENSAGEM (XX = FINAL) ";X$
140 IF X$ = "MESMA" THEN 200
150 IF X$ = "XX" THEN 270
160 L$ = X$
170 INPUT "QUANTOS BEEPS ";B
180 INPUT "LIMPAR A TELA (S/N) ";SN$
190 GOTO 210
200 X$ = L$
210 FOR I = 1 TO B
220 IF SN$ <> "S" THEN 240
230 CLS
240 PRINT I;TAB(6);X$;CHR$(7)
250 NEXT I
260 GOTO 120
270 CLS: END
```


ANIVERSARIANTES

O programa **Aniversariantes (ANIVER)** pode listar todos os nomes de seus conhecidos os quais fazem aniversário em qualquer mês ou fornece a você uma lista em ordem alfabética dos nomes e datas de aniversário.

Ele não é limitado à criação de lista de aniversariantes. Múltiplas cópias deste programa podem servir para diversos propósitos; além da versão para aniversariantes, uma versão com um calendário pessoal pode ser criada, ou uma combinação das duas. Por exemplo, **ANIVER** pode gerar um calendário de eventos para qualquer mês ou criar uma lista em ordem alfabética do nome dos eventos. Ou, se você quer ver todas as reuniões que você terá com uma determinada pessoa ou organização, você pode tirar uma listagem classificada (desde que, naturalmente, você seja cuidadoso ao inserir os nomes das reuniões da mesma maneira todas as vezes). Em 10 de janeiro, você pode ter dez coisas listadas para você lembrar, embora no dia 11, você tenha três; com este programa, você não tem que manter o registro de todas elas na sua memória.

Você pode substituir os comandos **DATA** nas linhas 130 - 480 com seus próprios nomes. Apenas digite seus próprios comandos **DATA** ao invés dos que estão litados (a menos que você queira lembrar da minha lista de aniversariantes). Note que o último comando **DATA** é 99/99/99 ZZZZZ (linha 480), ele é necessário para indicar ao comando **READ** da linha 510 que todos os dados de entrada foram lidos. Você pode acrescentar ou apagar dados de entrada através da digitação de novos comandos **DATA** ou da eliminação de alguns.

Vamos ver algumas instruções do programa que são de interesse. Após os comandos **DATA** terem sido lidos e colocados na matriz X$ nas linhas 500 - 530, suas opções são as seguintes:

1. Listar todos os aniversariantes ou reuniões em qualquer mês (01 a 12);
2. Listar em ordem alfabética todos os nomes ou reuniões descritas;
3. Finalizar o programa.

Estas opções são dadas nas linhas 560 - 710.

Se você digitar A (para uma listagem em ordem alfabética), o controle passa da linha 720 para a linha 1.150. O FIM determina que o programa seja finalizado e pula da linha 730 para a linha 1.480. As linhas 740 - 820 analizam qualquer outra seleção para ter certeza de que um número de dois dígitos foi digitado; elas olham em S1$ (o primeiro caracter) e S2$ (o segundo caracter) para ter certeza de que eles estão entre 0 e 9. O cheque não é muito perfeito, já que meses maiores do que 12 podem ser solicitados.

Se você digitar o número de um mês, o programa então classifica os dados, rearranjando-os em ordem, considerando os primeiros cinco caracteres (o dia, uma barra e o mês). Uma classificação do tipo "bolha" nas linhas 830 - 950 realiza

isto. Note que cada vez que você pedir as reuniões de um mês (ou outra informação), você reclassificará os dados. Você pode pensar que este processo é muito ineficiente; entretanto, se os dados já estão na ordem dia/mês, a classificação verifica o dado somente uma vez para verificar que os elementos já estão em seqüência. Lembre de outros exemplos do livro que na classificação pelo método da bolha, um sinalizador é mantido para ver se qualquer elemento está fora de ordem e então mudá-lo. Neste caso, o sinalizador é SW$ nas linhas 830, 930 e 950. Se nenhum elemento estiver fora de ordem, a classificação será terminada.

Depois, você é perguntado se quer uma listagem impressa ou exibida na tela (linha 980). Devido aos primeiros dois caracteres serem iguais ao mês selecionado, somente os dados deste mês são mostrados. (Veja a linha 1.010). Note que se você digitar 13 como o mês selecionado, todos os dados serão mostrados na seqüência dia/mês. Agora você pode ver que se digitar um número maior do que 13,o programa classificará (ou reclassificará) os dados na seqüência dia/mês — a menos que, naturalmente, você digite em seus comandos **DATA** meses maiores do que 12.

Se você selecionar uma lista em ordem alfabética, as linhas 1.160 - 1.300 classificam os dados em ordem alfabética, usando a chave de classificação a partir da décima posição do dado (após o espaço seguinte aos oito caracteres da data) para o final de cada elemento. A função MID$ nas linhas 1.220 - 1.230 analizam os nomes. Novamente, você pode obter uma listagem em ordem alfabética; se você não tiver uma impressora, poderá mostrar 20 ítens de cada vez na tela. Após cada tela, pressione **RETURN** para continuar.

```
10 REM ***********************
20 REM *                     *
30 REM *  ANIVERSARIANTES    *
40 REM *                     *
50 REM * DIREITOS AUTORAIS   *
60 REM *                     *
70 REM *   NELSON ALVES      *
80 REM *                     *
90 REM *                     *
100 REM **********************
110 KEY OFF
120 CLS
130 DATA 25/02/59 RUBENS
140 DATA 05/10/65 JOAO
150 DATA 01/05/29 ELIO
160 DATA 01/01/37 CLAUDINA
170 DATA 03/05/59 WILSON
```



```
180 DATA 03/05/56 TONY
190 DATA 15/03/62 LUCIANE
200 DATA 29/04/61 BERNARDETE
210 DATA 07/05/57 CRISTINA
220 DATA 29/05/81 ERIKA
230 DATA 23/12/78 SISSI
240 DATA 09/04/33 AGOSTINHO
250 DATA 24/02/59 NELSON
260 DATA 09/09/11 PASCOALATO
270 DATA 24/05/59 ELIANE
280 DATA 06/22/67 PEDRO
290 DATA 25/06/45 CARLOS
300 DATA 03/01/71 ROBERTO
310 DATA 09/08/45 DEBORA
320 DATA 19/11/18 ANDREA
330 DATA 15/04/71 CRIS
340 DATA 29/08/78 HENRIQUE
350 DATA 18/08/73 JOSE
360 DATA 23/03/69 PAULO
370 DATA 07/05/53 OSVALDO
380 DATA 31/10/58 ANTENOR
390 DATA 19/02/41 MARCOS
400 DATA 26/10/70 AMAURI
410 DATA 15/05/40 PORCINA
420 DATA 20/09/30 MALTA
430 DATA 09/11/45 FABIO
440 DATA 14/02/47 MATILDE
450 DATA 30/03/59 ISABEL
460 DATA 24/12/43 ARTHUR
470 DATA 13/08/13 ASTROMAR
480 DATA 99/99/99 ZZZZZ
490 DIM X$(200)
500 FOR I = 1 TO 200
510 READ X$(I)
520 IF MID$ (X$(I),1,2) = "99" THEN 540
530 NEXT I
540 CTR = I
550 CLS
```

```
560 PRINT "***********************************"
570 PRINT "*                                 *"
580 PRINT "*        ANIVERSARIOS NO MES      *"
590 PRINT "*                                 *"
600 PRINT "*                                 *"
610 PRINT "*                                 *"
620 PRINT "*   (13 = TODOS OS MESES)         *"
630 PRINT "*                                 *"
640 PRINT "*   (A  = ANIVERSARIANTES EM ORDEM) *"
650 PRINT "*                                 *"
660 PRINT "*   (FIM= FINAL DO PROGRAMA)      *"
670 PRINT "*                                 *"
680 PRINT "*                                 *"
690 PRINT "***********************************"
700 PRINT:PRINT:PRINT
710 INPUT "SELECAO ";S$
720 IF S$ = "A" THEN 1150
730 IF S$ = "FIM" THEN 1480
740 S1$ = MID$(S$,1,1)
750 S2$ = MID$(S$,2,1)
760 IF S1$ < "0" THEN 550
770 IF S1$ > "9" THEN 550
780 IF S2$ < "0" THEN 550
790 IF S2$ > "9" THEN 550
800 M$ = S1$ + S2$
810 PASS = 0
820 IF LEN(M$) <> 2 THEN 550
830 SW$ = "N"
840 PASS = PASS+1
850 CLS
860 PRINT
870 PRINT"REALIZANDO O SORT: ";PASS
880 FOR I =1 TO CTR-1
890 IF MID$(X$(I),1,5) <= MID$(X$(I+1),1,5) THEN 940
900 Y$ = X$(I)
910 X$(I) = X$(I+1)
920 X$(I+1) = Y$
930 SW$ = "S"
940 NEXT I
950 IF SW$ = "S" THEN 830
960 CX = 0
970 PRINT
980 INPUT "IMPRIMIR (S/N) ";ISN$
990 FOR I = 1 TO CTR
1000 IF M$ = "13" THEN 1020
```



```
1010 IF MID$(X$(I),4,2) <> M$ THEN 1110
1020 IF ISN$ = "S" THEN 1050
1030 PRINT X$(I)
1040 IF ISN$ <> "S" THEN 1060
1050 LPRINT X$(I)
1060 CX = CX+1
1070 IF CX < 20 THEN 1110
1080 CX = 0
1090 PRINT
1100 INPUT "PRESSIONE RETURN PARA CONTINUAR ";E$
1110 NEXT I
1120 PRINT
1130 INPUT "PRESSIONE RETURN PARA CONTINUAR ";E$
1140 GOTO 550
1150 PASS = 0
1160 SW$ = "N"
1170 PASS = PASS+1
1180 CLS
1190 PRINT
1200 PRINT "REALIZANDO O SORT: ";PASS
1210 FOR I = 1 TO CTR-1
1220 S1$ = MID$ (X$(I),10)
1230 S2$ = MID$(X$(I+1),10)
1240 IF S1$ < S2$ THEN 1290
1250 SW$ = "S"
1260 Y$ = X$(I)
1270 X$(I) = X$(I+1)
1280 X$(I+1) = Y$
1290 NEXT I
1300 IF SW$ = "S" THEN 1160
1310 CX = 0
1320 PRINT
1330 INPUT "IMPRIMIR (S/N) ";ISN$
1340 FOR I = 1 TO CTR
1350 IF ISN$ = "S" THEN 1380
1360 PRINT X$(I)
1370 IF ISN$ <> "S" THEN 1390
1380 LPRINT X$(I)
1390 CX = CX+1
1400 IF CX < 20 THEN 1440
1410 CX = 0
1420 PRINT
1430 INPUT "PRESSIONE RETURN PARA CONTINUAR ";E$
1440 NEXT I
1450 PRINT
```

```
1460 INPUT "PRESIONE RETURN PARA CONTINUAR ";E$
1470 GOTO 550
1480 CLS : END
```

```
01/01/37 CLAUDINA
01/05/29 ELIO
03/01/71 ROBERTO
03/05/59 WILSON
03/05/56 TONY
05/10/65 JOAO
06/22/67 PEDRO
07/05/57 CRISTINA
07/05/53 OSVALDO
09/04/33 AGOSTINHO
09/08/45 DEBORA
09/09/11 PASCOALATO
09/11/45 FABIO
13/08/13 ASTROMAR
14/02/47 MATILDE
15/03/62 LUCIANE
15/04/71 CRIS
15/05/40 PORCINA
18/08/73 JOSE
19/02/41 MARCOS
19/11/18 ANDREA
20/09/30 MALTA
23/03/69 PAULO
23/12/78 SISSI
24/02/59 NELSON
24/05/59 ELIANE
24/12/43 ARTHUR
25/02/59 RUBENS
25/06/45 CARLOS
26/10/70 AMAURI
29/04/61 BERNARDETE
29/05/81 ERIKA
29/08/78 HENRIQUE
30/03/59 ISABEL
31/10/58 ANTENOR
99/99/99 ZZZZZ
```

Figura 3. Programa Aniversariantes: exemplo de saída do programa.

BLACKJAC

O programa **BlackJac (BLAJA)** permite que você participe de uma partida de BLACKJACK (ou Vinte-e-Um) contra o seu MSX, o qual age como se fosse a mesa. Você e a mesa recebem duas cartas cada; as duas faces das suas cartas estão para cima, mas uma carta da mesa está com a face para baixo. À medida que seus pontos estejam abaixo de vinte-e-um, é possível pedir tantas cartas quanto você queira; caso contrário você "estoura" e perde. As cartas de 2 a 10 têm valores de 2 - 10; o Valete, a Dama e o Rei tem valor 10 cada; um Ás tem valor 1 ou 11. Para pedir uma carta, simplesmente responda S à pergunta "PRESSIONE (S OU N)" (linhas 790 - 800). Você não precisa pressionar **RETURN** porque a instrução, **INKEY$**, aceita o caracter de uma tecla apenas.

As linhas 130 - 160 inicializam as matrizes que mantêm registro das cartas da mesa (MESA), suas cartas (CARTA), os naipes (S$), e a denominação (D$). O *loop* nas linhas 220 - 240 lê as cartas de "A" a "K" do comando **DATA** na linha 250 e as coloca na matriz D$. Nas linhas 260 - 270, a função definida pelo usuário FN I é criada para pedir uma carta aleatoriamente da mesa; é dado a você uma quantia de 100 cruzados para apostas.

No início de cada jogo, as linhas 290 - 320 colocam em 0 os valores de todas as cartas que a mesa e você tinham, e as linhas 330 - 350 reembaralham as cartas de modo que qualquer carta esteja disponível para distribuição. O seu total — os pontos que as cartas irão somar ao seu escore — é colocado em 0, como as da mesa também.

A subrotina **GOSUB** 1.960 produz o cabeçalho "BLACKJACK " — com asteriscos em ambos os lados. Você fica sabendo na linha 400 quanto dinheiro tem disponível. A linha 410 pergunta pela quantia que você quer apostar. Para sair, digite 0 ou um número negativo. As linhas 430 - 550 distribuem as quatro primeiras cartas, aleatoriamente escolhidas como o primeiro e o segundo subscritos de CARTA e MESA: CARTA(1), CARTA(2), MESA(1) e MESA(2). A função FN I (descrita na linha 260) escolhe estes subscritos; as linhas 600 - 770 mostram os valores das cartas. Para mostrá-los, o programa define um valor para a variável CARTNO (de 1 a 52) para carta selecionada; a subrotina nas linhas 1.270 - 1.360 obtém o naipe e a denominação correta. As cartas são então mostradas, mas a segunda carta da mesa é deixada de face para baixo (representada pelos dois pontos de interrogação). Na linha 780, é dado o valor 2 à variável PART de modo que a próxima jogada que você ficar será com a carta número 3, e assim por diante.

As linhas 790 - 800 perguntam se você quer outra carta. As linhas 850 - 890 procuram outra carta, e o comando **GOSUB** da linha 1.370 calcula seu escore total. A linha 900 então verifica para ver se você "estourou", neste caso o programa salta para a linha 1.030 para dizer que você perdeu e mostra o seu escore. Neste ponto, você pode tanto pressionar 1 para finalizar ou qualquer outra tecla para continuar (linhas 1.100 - 1.110). Se escolher parar, as linhas 920 -

1.010 permitem à mesa (o computador) pedir cartas até ela ter ganho ou estourado. A subrotina **GOSUB** 1.660 na linha 930 obtém os valores das cartas pedidas e mostra-as na tela. As linhas 1.020 - 1.250 novamente mostram as mensagens do vencedor ou perdedor, e passam o controle da linha 1.140 de volta para a linha 290 para o próximo jogo. Experimente. É um jogo para todas as idades!

```
10 REM  ***********************
20 REM  *                     *
30 REM  *     BLACK JACK      *
40 REM  *                     *
50 REM  *  DIREITOS AUTORAIS  *
60 REM  *                     *
70 REM  *    NELSON ALVES     *
80 REM  *                     *
90 REM  *                     *
100 REM ***********************
110 KEY OFF
120 CLS
130 DIM MESA(20)
140 DIM CARTA(20)
150 DIM S$(4)
160 DIM D$(13)
170 REM AS CARTAS SAO:
180 REM 0 = PAUS
190 REM 1 = OUROS
200 REM 2 = COPAS
210 REM 3 = ESPADAS
220 FOR I =1 TO 13
230 READ D$(I)
240 NEXT I
250 DATA A,2,3,4,5,6,7,8,9,10,J,Q,K
260 DEF FN I(X) = INT(RND(X)*52)+1
270 APT = 100
280 DIM DRWN(52)
290 FOR I = 0 TO 20
300 MESA(I) = 0
310 CARTA(I) = 0
```



```
320 NEXT I
330 FOR I = 1 TO 52
340 DRWN(I) = 0
350 NEXT I
360 JOG = 0: VOCE = 0
370 CLS
380 GOSUB 1960
390 LOCATE 1,5
400 PRINT "O VALOR DA APOSTA E DE: ";APT
410 PRINT:INPUT "VOCE QUER APOSTAR (0 FINALIZA) ";AJOG
420 IF AJOG < 1 OR AJOG > APT THEN 1990
430 REM *** PRIMEIRAS 4 CARTAS ***
440 CARTA(1) = FN I(1)
450 IF DRWN(CARTA(1)) = 1 THEN 440
460 DRWN(CARTA(1)) = 1
470 CARTA(2) = FN I(1)
480 IF DRWN(CARTA(2)) = 1 THEN 470
490 DRWN(CARTA(2)) = 1
500 MESA(1) = FN I(1)
510 IF DRWN(MESA(1)) = 1 THEN 500
520 DRWN(MESA(1)) = 1
530 MESA(2) = FN I(1)
540 IF DRWN(MESA(2)) = 1 THEN 530
550 DRWN(MESA(2)) = 1
560 REM *** MOSTRA AS CARTAS ***
570 CLS
580 GOSUB 1950
590 CARTNO = MESA(1)
600 REM *** MOSTRA O VALOR DA CARTA ***
610 GOSUB 1270
620 LOCATE 1,5
630 PRINT "MESA"
640 LOCATE 8,5
650 PRINT DISP$
660 LOCATE 12,5
670 PRINT "??"
680 CARTNO = CARTA(1)
690 GOSUB 1270
700 LOCATE 1,8
710 PRINT "VOCE"
720 LOCATE 8,8
730 PRINT DISP$
740 CARTNO = CARTA(2)
750 GOSUB 1270
760 LOCATE 12,8
```

```
770 PRINT DISP$
780 PART = 2
790 LOCATE 5,20
800 PRINT "PRESSIONE (S OU N) "
810 K$ = INKEY$
820 IF K$ = "" THEN 810
830 IF K$ <> "S" THEN GOSUB 1370
840 IF K$ <> "S" THEN 920
850 PART = PART+1
860 CARTA(PART) = FN I(1)
870 IF DRWN(CARTA(PART)) = 1 THEN 860
880 DRWN(CARTA(PART)) = 1
890 GOSUB 1370
900 IF BUST$ = "S" THEN 1020
910 GOTO 790
920 PART = 2
930 GOSUB 1660
940 IF BD$ = "S" THEN 1190
950 IF VOCE = 21 THEN 1150
960 IF JOG >= VOCE AND JOG <> 0 THEN 1230
970 PART = PART+1
980 MESA(PART) = FN I(1)
990 IF DRWN(MESA(PART)) = 1 THEN 980
1000 DRWN(MESA(PART)) = 1
1010 GOTO 930
1020 LOCATE 1,15
1030 PRINT: PRINT: PRINT "VOCE PERDEU !!"
1040 APT = APT - AJOG
1050 PRINT: PRINT
1060 PRINT "MESA: ";JOG;SPC(5);"VOCE: ";VOCE
1070 IF APT > 0 THEN 1100
1080 FOR I = 1 TO 2000:NEXT I: GOTO 1990
1090 CLS: LOCATE 0,10: PRINT "O JOGO ACABOU, POIS VOCE
1100 PRINT:PRINT "RETURN = CONTINUAR, 1 = PARAR"
1110 K$ = INKEY$
1120 IF K$ = "" THEN 1110
1130 IF K$ = "1" THEN 1990
1140 GOTO 290
1150 LOCATE 1,15
1160 PRINT "VOCE GANHOU - VINTE-E-UM !!!!";CHR$(7)
1170 APT = APT + AJOG
1180 GOTO 1050
1190 LOCATE 1,15
1200 PRINT "VOCE GANHOU - A MESA ESTOUROU !";CHR$(7)
1210 APT = APT + AJOG
```



```
1220 GOTO 1050
1230 LOCATE 1,15
1240 PRINT "VOCE PERDEU - CARTA MENOR OU IGUAL !"
1250 GOTO 1040
1260 REM *** LE CARTA E DENOMINACAO ***
1270 NAIPE = INT(CARTNO/13)
1280 IF CARTNO = 13*NAIPE THEN NAIPE = NAIPE-1: S2 = NAIPE
1290 DENOM = CARTNO - (NAIPE*13)
1300 ON NAIPE+1 GOTO 1310,1320,1330,1340
1310 S$ = "P": GOTO 1350
1320 S$ = "O": GOTO 1350
1330 S$ = "C": GOTO 1350
1340 S$ = "E"
1350 DISP$ = D$(DENOM)+S$
1360 RETURN
1370 Y1 = 0
1380 AZ = 0
1390 Y2 = 0
1400 FOR I =1 TO 20
1410 IF CARTA(I)= 0 THEN 1580
1420 NAIPE = INT(CARTA(I)/13)
1430 IF CARTA(I) = 13*NAIPE THEN NAIPE = NAIPE-1: S2 = NAIPE
1440 DENOM = CARTA(I)- (NAIPE*13)
1450 ON NAIPE GOTO 1460,1470,1480,1490
1460 S$ = "P": GOTO 1500
1470 S$ = "O": GOTO 1500
1480 S$ = "C": GOTO 1500
1490 S$ = "E"
1500 DISP$ = D$(DENOM)+S$
1510 IF DENOM > 10 THEN DENOM = 10
1520 IF DENOM = 1 AND AZ = 0 THEN Y2 = Y2+11
1530 IF DENOM = 1 AND AZ = 1 THEN Y2 = Y2 + DENOM
1540 IF DENOM <> 1 THEN Y2 = Y2 + DENOM
1550 IF DENOM = 1 AND AZ = 0 THEN AZ = 1
1560 Y1 = Y1 + DENOM
1570 NEXT I
1580 BUST$ = CHR$(32)
1590 IF Y1 > 21 AND Y2 > 21 THEN BUST$ = "S"
1600 VOCE = Y2
1610 IF VOCE > 21 THEN VOCE = Y1
1620 CARTNO = CARTA(PART): GOSUB 1270
1630 LOCATE 8+4*(PART-1),8
1640 PRINT DISP$
1650 RETURN
1660 REM *** A MESA JOGA ***
```

```
1670 D1 = 0
1680 AZ = 0
1690 D2 = 0
1700 FOR I = 1 TO 20
1710 IF MESA(I) = 0 THEN 1880
1720 NAIPE = INT(MESA(I)/13)
1730 IF MESA(I) = 13*NAIPE THEN NAIPE = NAIPE-1 : S2 = NAIPE
1740 DENOM = MESA(I) - (NAIPE*13)
1750 ON NAIPE+1 GOTO 1760,1770,1780,1790
1760 S$ = "P": GOTO 1800
1770 S$ = "O": GOTO 1800
1780 S$ = "C": GOTO 1800
1790 S$ = "E"
1800 DISP$ = D$(DENOM) + S$
1810 IF DENOM > 10 THEN DENOM = 10
1820 IF DENOM = 1 AND AZ = 0 THEN D2 = D2 + 11
1830 IF DENOM = 1 AND AZ = 1 THEN D2 = D2 + DENOM
1840 IF DENOM <> 1 THEN D2 = D2 + DENOM
1850 IF DENOM = 1 AND AZ = 0 THEN AZ = 1
1860 D1 = D1 + DENOM
1870 NEXT I
1880 BD$ = CHR$(32)
1890 IF D1 > 21 AND D2 > 21 THEN BD$ = "S"
1900 JOG = D2
1910 IF JOG > 21 THEN JOG = D1
1920 CARTNO = MESA(PART): GOSUB 1270
1930 LOCATE 8+4*(PART-1),5
1940 PRINT DISP$
1950 RETURN
1960 LOCATE 3,2
1970 PRINT "* * * * *  BLACK JACK  * * * * *"
1980 RETURN
1990 CLS:END
```


DADOS

O programa **Dados (JOGO)** é um jogo feito para até 10 jogadores. As regras são muito simples: todos os jogadores iniciam com um montante de 100 cruzados ou pontos (linhas 380 - 400). No início do programa, o limite da casa é acertado para qualquer aposta; caso contrário, perdedores podem continuar apostando cada vez mais até finalmente ganharem uma partida.

O primeiro jogador faz sua aposta e rola os dados. Se a primeira jogada for um 7 ou um 11, este jogador ganha e o próximo jogador tem a sua vez. Se a primeira jogada for um 2, um 3, ou um 12, o jogador perde. Caso contrário, o total da primeira jogada torna-se os pontos do jogador. Ele joga continuamente até obter os pontos ou tirar um 7 e perder.

Os escores são mantidos na matriz S, a qual é definida na linha 120. O ajuste de K para "N" na linha 130 faz com que o programa desconsidere os escores dos jogadores (de um jogo anterior, por exemplo). O "NO> DE MISTURADAS DESEJADO" caracterizado na linha 690 sempre garante que ninguém será capaz de predeterminar a saída dos dados rolados consistentemente. Você pode misturar os dados em sua mão qualquer número de vezes; mas,obviamente, isto não deve ser feito um número muito grande de vezes.

As linhas 150 - 290 fornecem uma breve descrição na tela, de como o jogo funciona. Se você já está familiarizado com as instruções, pode pressionar **RETURN** para prosseguir. As linhas 300 - 400 então realizam certas rotinas de inicialização (processamento de dados para inicialização de variáveis de início de programa). Você pode inicializar o limite de apostas da mesa, o número de jogadores, e o número de partidas que cada jogador terá. Quando este limite for atingido, o jogo termina; mas você pode escolher jogar novamente. Pode também manter os escores dos jogos anteriores respondendo S para a pergunta da linha 1.000 (a qual coloca a variável K em "S").

As linhas 410 - 940 contêm um grande *loop* que é executado tantas vezes quanto o número de partidas que foi selecionado. Com ele, o *loop* **FOR-NEXT** das linhas 420 - 930 deixa cada jogador ter a sua vez durante cada partida. Na linha 430, CTR (representando o número de jogadas feitas) e a variável ULT são colocadas em 0. A variável ULT, que representa o ponteiro do escore que você está tentando igualar, é ajustada para o valor da sua primeira jogada (linha 870).

Durante o tempo em que o primeiro jogador estiver jogando, o programa saltará para a linha 430 para deixar o jogador jogar novamente os dados. Neste caso, se você não obtiver uma vitória nem for derrotado na primeira jogada, a linha 900 saltará para a linha 440 para a sua segunda tentativa. Do mesmo modo, se você não conseguir seus pontos e não tirar um sete, a linha 840 retornará para a linha 440 para a sua próxima jogada. Você deve jogar até ganhar ou perder.

As linhas 540 - 680 perguntam pela aposta de cada jogador — que não pode exceder o limite da mesa — e avisa se o jogador não tem bastante dinheiro ou

pontos para cobrir a aposta (embora a aposta seja aceita de qualquer forma). A seguir o programa pergunta quantas vezes você quer misturar os dados. Seus dados são rolados nas linhas 710 - 740, e a subrotina chamada da linha 760 imprime os números que saíram. Se esta for a primeira jogada (determinada pelo valor de CTR), a linha 770 saltará para a linha 850 para uma lógica especial para a primeira jogada. Na segunda jogada e assim sucessivamente, as linhas 780 - 840 determinam se você ganhou através dos pontos (linha 780), perdeu tirando um 7 (linha 790), ou deve jogar novamente (linha 810). Na primeira jogada, as linhas 850 e 860 dão a você a vitória ou a derrota, enquanto a linha 870 dá o total dos seus dados (DT) para ULT (o ponteiro que indica que você deve jogar). A cada vez que alguém jogar, a rotina nas linhas 1.030 - 1.180 imprimirão as quantidades de dinheiro ou pontos restantes para cada jogador. Estas quantidades são também mostradas no final do programa. (Veja linhas 520 e 960).

```
10 REM  **********************
20 REM  *                    *
30 REM  *       JOGO         *
40 REM  *                    *
50 REM  *  DIREITOS AUTORAIS *
60 REM  *                    *
70 REM  *    NELSON ALVES    *
80 REM  *                    *
90 REM  *                    *
100 REM **********************
110 KEY OFF
120 DIM S(10)
130 K$ = "N"
140 CLS
150 LOCATE 13,5
160 PRINT "*** JOGO ***"
170 PRINT
180 PRINT "ESTE JOGO PODE TER ATE 10 JOGADORES."
190 PRINT "ELE E BASEADO NO JOGO REAL DE DADOS."
200 PRINT "VOCE PODE JOGAR QUANTAS VEZES QUISER, "
210 PRINT "O JOGADOR GANHA SE ELE TIRAR UM 7 OU"
220 PRINT "UM 11 NA PRIMEIRA JOGADA."
230 PRINT "SE TIRAR UM 2, 3 OU 12 NA PRIMEIRA"
240 PRINT "JOGADA, ELE PERDE.  QUALQUER OUTRO "
250 PRINT "NO. NA PRIMEIRA JOGADA E O PONTO QUE "
260 PRINT "ELE DEVE TIRAR NOVAMENTE ANTES DE "
270 PRINT "TIRAR UM 7."
280 LOCATE 0,23
```



```
290 INPUT "PRESSIONE RETURN PARA CONTINUAR ";X$
300 CLS
310 LOCATE 0,5
320 INPUT "LIMITE DE APOSTA DA CASA ";LIMIT
330 PRINT
340 INPUT "NUMERO DE JOGADORES ";N1
350 PRINT
360 INPUT "NUMERO DE JOGADAS ";R1
370 IF K$ = "S" THEN 410
380 FOR I = 1 TO N1
390 S(I) = 100
400 NEXT I
410 FOR PART = 1 TO R1
420 FOR JOG = 1 TO N1
430 CTR = 0: ULT = 0
440 CTR = CTR + 1
450 CLS
460 PRINT "JOGADA... ";PART
470 PRINT "JOGADOR NO. ";JOG
480 PRINT "ROLOU";SPC(1);CTR;SPC(1);"VEZ(ES)"
490 IF CTR = 1 THEN 510
500 PRINT "PONTO TIRADO: ";ULT
510 PRINT
520 GOSUB 1030
530 IF CTR <> 1 THEN 680
540 LOCATE 0,10
550 INPUT "VALOR DA APOSTA ";VPT
560 IF VPT <= LIMIT THEN 640
570 PRINT "O LIMITE DA CASA E Cz$ ";LIMIT
580 PRINT:INPUT "PRESSIONE RETURN ";X$
590 LOCATE 0,10
600 FOR I = 1 TO 3
610 PRINT SPC(53)
620 NEXT I
630 GOTO 540
640 IF AMT <= S(JOG) THEN 680
650 LOCATE 0,10
660 PRINT "VOCE TEM SOMENTE Cz$ ";S(JOG)
670 PRINT: INPUT "PRESSIONE RETURN ";X$
680 LOCATE 0,12
690 PRINT:INPUT "NO. DE MISTURADAS DESEJADO ";SH
700 IF SH < 1 THEN SH = 1
710 FOR J = 1 TO SH
720 D1 = INT(6*RND(1))+1
730 D2 = INT(6*RND(1))+1
```

```
740 NEXT J
750 DT = D1 + D2
760 GOSUB 1180
770 IF CTR = 1 THEN 850
780 IF DT=ULT THEN LOCATE 20,15:PRINT "VOCE GANHOU!";
    CHR$(7):S(JOG)=S(JOG)+VPT:GOTO 910
790 IF DT=7 THEN LOCATE 20,15:PRINT "VOCE PERDEU !";
    CHR$(7):S(JOG)=S(JOG)-VPT:GOTO 910
800 LOCATE 0,15
810 PRINT "TENTE NOVAMENTE, SEU PONTO E ";ULT
820 LOCATE 15,20
830 INPUT "PRESSIONE RETURN ";X$
840 GOTO 440
850 IF DT=7 OR DT=11 THEN LOCATE 20,15:PRINT "VOCE VENCEU !";
    CHR$(7):S(JOG)=S(JOG)+VPT:GOTO 910
860 IF DT=2 OR DT=3 OR DT=12 THEN LOCATE 20,15:
    PRINT "VOCE PERDEU !";CHR$(7):S(JOG)=S(JOG)-VPT:GOTO 910
870 ULT = DT
880 LOCATE 15,20
890 INPUT "PRESSIONE RETURN ";X$
900 GOTO 440
910 LOCATE 15,20
920 INPUT "PRESSIONE RETURN ";X$
930 NEXT JOG
940 NEXT PART
950 CLS
960 GOSUB 1030
970 LOCATE 0,15
980 INPUT "JOGAR NOVAMENTE (S/N) ";IS$
990 IF IS$ = "N" THEN 1020
1000 INPUT "MANTER A APOSTA (S/N) ";K$
1010 GOTO 140
1020 CLS: END
1030 PRINT "DINHEIRO RESTANTE: ";
1040 PRINT
1050 FOR I = 1 TO 4
1060 IF I > N1 THEN 1170
1070 PRINT "#";I;SPC(1);"Cz$";S(I);SPC(2);CHR$(207);
1080 NEXT I
1090 FOR I = 5 TO 8
1100 IF I > N1 THEN 1170
1110 PRINT "#";I;SPC(1);"CZ$";S(I);SPC(2);CHR$(207);
1120 NEXT I
1130 FOR I = 9 TO 10
1140 IF I > N1 THEN 1170
```



```
1150 PRINT "#";I;SPC(1);"Cz$";S(I);SPC(2);CHR$(207);
1160 NEXT I
1170 RETURN
1180 PRINT
1190 PRINT D1;SPC(1);"+";SPC(1);D2;SPC(1);"=";SPC(1);DT
1200 RETURN
```

DESENHO

O programa **DESENHO (DESENHO)** permite usar blocos ou asteriscos para desenhar qualquer figura ou diagrama que queiramos na tela. Se você primeiro desenhar o *layout* de sua figura ou gráfico no papel, preenchendo alguns pontos e deixando outros em branco, pode então duplicá-lo na tela.

Sua figura pode ter até 20 linhas (para cima e para baixo) e 30 colunas de comprimento (através da tela). A linha 150 pergunta se você quer usar blocos ou astericos para desenhar o seu diagrama; você deve responder com "*" ou B.

A seguir você precisa dizer ao seu computador MSX que colunas em cada linha estão preenchidas. O programa iniciará com a LINHA 1 e a COLUNA 1. Você deve digitar a primeira coluna (um número entre 1 e 30) na qual você quer um bloco. A seguir você é perguntado pela LINHA 1, POSIÇÃO 2, para a qual você digita a segunda coluna onde quer um círculo. Este processo é repetido para a LINHA 1 até que você digite 0 (zero) para avançar para a próxima linha, 998 finaliza o programa, ou 999 para desenhar a figura.

O desenho atual toma forma nas linhas 310 - 550. Você é perguntado na linha 310 quantas vezes quer desenhar a figura, apagá-la e então redesenhá-la. Se você selecionar os blocos, os comandos LOCATE da linha 400 e o comando CHR$ da linha 410 colocam os blocos nas linhas e colunas apropriadas. (Veja o seu manual do BASIC-MSX para maiores detalhes). Ou se você selecionar asteriscos, impressões nas colunas são utilizadas.

Quando o número de repetições que você requisitou for mostrado, você poderá desenhar o mesmo diagrama (linha 530), ou pode criar um novo.

```
10 REM  ***********************
20 REM  *                     *
30 REM  *       DESENHO       *
40 REM  *                     *
50 REM  *  DIREITOS AUTORAIS  *
60 REM  *                     *
70 REM  *    NELSON ALVES     *
80 REM  *                     *
90 REM  *                     *
100 REM ***********************
110 KEY OFF
120 DIM X(20,30)
130 CLS
140 PRINT
150 INPUT "* OU B PARA O BLOCO ";A$
```



```
160 FOR I = 1 TO 20
170 FOR J = 1 TO 30
180 PRINT "LINHA";I;"POSICAO";J
190 PRINT "(0=AVANCA PARA A PROXIMA LINHA)"
200 PRINT "(999=DESENHA O DIAGRAMA)"
210 PRINT "(998=FINAL DO PROGRAMA)"
220 PRINT "(##=COLUNA DA LINHA EM USO)"
230 INPUT N
240 X(I,J)=N
250 IF N = 0 THEN 290
260 IF N = 999 THEN 300
270 IF N = 998 THEN 560
280 NEXT J
290 NEXT I
300 CLS
310 INPUT "NO. DE VEZES ";N2
320 N3 = 0
330 CLS
340 FOR I = 1 TO 20
350 FOR J = 1 TO 30
360 IF X(I,J) = 0 THEN 440
370 IF X(I,J) = 999 THEN 480
380 N = X(I,J)
390 IF A$ <> "B" THEN 430
400 LOCATE N,I
410 PRINT CHR$(219);
420 GOTO 440
430 PRINT TAB(N);"*";
440 NEXT J
450 IF A$ = "B" THEN 470
460 PRINT
470 NEXT I
480 IF A$ = "B" THEN 500
490 PRINT: PRINT: PRINT
500 N3 = N3 + 1
510 IF N3 >= N2 THEN 530
520 GOTO 330
530 LOCATE 10,20: INPUT "REPETIR (S/N) ";SN$
540 IF SN$ = "S" THEN 300
550 GOTO 130
560 PRINT: PRINT: PRINT
570 CLS: END
```

EDIÇÃO DE NÚMEROS

O programa de **Edição de Números (EDITOR)** é um programa bastante simples que toma qualquer número digitado (por exemplo 1200050) e edita-o, colocando pontos e uma vírgula para a casa dos centavos, e/ou um sinal de menos, se necessário. Por exemplo, o resultado do número acima seria 12.000,50.

O EDITOR é importante no que ele demonstra como você pode obter um resultado formatado, usando um BASIC comum que não tem a opção do comando **PRINT USING.** Com esta subrotina qualquer programa pode pegar um número, assim como N1 (linha 130); colocá-lo na variável **NUMED** (linha (180); fornecer uma máscara de edição assim como M$ (linha 190); e obter o resultado assim pela subrotina **GOSUB** 230 que realiza a subrotina na linha 230 até o comando **RETURN** ser encontrado.

A subrotina procura da direita para a esquerda através da máscara de edição e substitui os dígitos por números da variável NUMED. Zeros à esquerda são desprezados, assim como pontos desnecessários, para obter o resultado.

```
10 REM  ***********************
20 REM  *                     *
30 REM  *  EDICAO DE NUMEROS  *
40 REM  *                     *
50 REM  *  DIREITOS AUTORAIS  *
60 REM  *                     *
70 REM  *     NELSON ALVES    *
80 REM  *                     *
90 REM  *                     *
100 REM ***********************
110 KEY OFF
120 CLS
130 INPUT "NO. A SER EDITADO ";N1
140 IF N1 = INT(N1) THEN 170
150 PRINT "DEVE SER UM NUMERO INTEIRO!"
160 GOTO 120
170 IF N1 = 9 THEN 580
180 NUMED = N1
190 M$ = "XX.XXX.XXX,XX-"
200 GOSUB 230
210 PRINT M$
220 GOTO 130
230 REM *** SUBROTINA DE EDICAO ***
240 REM *** MASC$ = MASCARA DE EDICAO ***
250 REM *** EXEMPLO: XX.XXX.XXX,XX- ***
```



```
260 REM *** NUMED = NO. PARA EDITAR (INTEIRO) ***
270 N2 = NUMED
280 NUMED = ABS(NUMED)
290 LM = LEN(M$)
300 NUMED$ = STR$(NUMED)
310 LNUMED = LEN(NUMED$)
320 CN = LNUMED + 1
330 FOR CTM = LM TO 1 STEP -1
340 CN = CN - 1
350 IF CN < 1 THEN 530
360 IF MID$(MC$,CTM,1) <> "-" THEN 440
370 IF N2 > 0 THEN 410
380 MASC$ = MID$(MASC$,1,CTM-1) + "-"
390 CN = CN + 1
400 GOTO 430
410 MASC$ = MID$(MASC$,1,CTM-1) + CHR$(32)
420 CN = CN + 1
430 GOTO 540
440 IF MID$(M$,CTM,1) = "," THEN 470
450 IF MID$(M$,CTM,1) = "." THEN 470
460 GOTO 500
470 CN = CN + 1
480 IF CTM = 2 THEN CN = CN - 1
490 GOTO 540
500 M1$=MID$(M$,1,CTM-1)+MID$(NUMED$,CN,1)+
    MID$(M$,CTM+1,LM-(CTM-1)-1)
510 M$ = M1$
520 GOTO 540
530 M$=MID$(M$,1,CTM-1)+CHR$(32)+
    MID$(M$,CTM+1,(LM-(CTM-1)-1))
540 REM *** IMPRIME CTM;MASC$ ***
550 REM PRINT M$,CTM,CN:STOP
560 NEXT CTM
570 RETURN
580 CLS: END
```

JOGO DE DADOS

O programa **Jogo de Dados (JOGO)** permite que até 10 jogadores participem. As instruções nas linhas 250 - 390 sumarizam rapidamente as regras do jogo.

Cada partida do jogo começa na linha 560. As linhas 440 - 510 pedem o nome dos jogadores (especificados na linha 480) a partir da matriz P$, e um *loop* FOR-NEXT (linhas 570 - 1.580) dando a vez de cada jogador. Quando todos os jogadores tiverem jogado na primeira partida, o controle será passado da linha 1590 para a linha 560, onde o número de partidas é incrementado e outro *loop* **FOR-NEXT** começará. Este processo continua até um dos jogadores ter um escore igual ao número de pontos selecionados na linha 280.

Note que após você misturar os dois dados, se escolheu misturar de novo ambos os dados em uma tentativa de obter o dobro — e falhou, seu escore será zero para esta tentativa. As linhas 600 - 720, usando a seleção de números aleatórios entre 1 e 6, obtêm o resultado original da jogada dos dois dados.

As linhas 840 - 1.020 permitem que você jogue o dado 1 novamente em uma tentativa de obter o dobro e a linha 790 analiza se este dado é menor que, ou igual, ao outro dado. Se você já tem o dobro, mas decidir jogar novamente, terá uma chance em duas de tirar o mesmo dobro. Você deve simplesmente responder com N na linha 730 se tiver o dobro. (Lembre-se de que você recebe 10 pontos por qualquer dobro cujo ponto total vezes dois é menor do que 10). As linhas 1.030 - 1.250 fazem o mesmo procedimento para a tentativa feita com o dado 2.

A única coisa sobre este jogo é que ele não pode ser duplicdo "como está" sem um computador. Para ter uma chance em duas de tirar o dobro com um dado real, você necessitará de três oportunidades para tentar igualar com o outro dado; similarmente, para ter uma chance em três, necessitará duas jogadas.

Com JOGO, entretanto, você tem somente que dizer ao programa que deseja jogar novamente o dado1 ou o dado2 (linha 730), ou responder N para declinar da tentativa. Então o programa automaticamente determina se você está jogando o dado maior ou o dado menor e computa o resultado, dando a você os ímpares apropriados (linhas 840 e 940). Se você jogar novamente o dado menor, a linha 840 selecionará um número "randômico" (aleatório) entre 1 e 3. Se o número 1 for selecionado, você terá o dobro. A mesma lógica aplica-se para uma chance em duas, exceto que você tem o resultado de um número "randômico" entre 1 e 2.

Seu escore será mostrado nas linhas 1.270 - 1.510. As linhas 1.330 - 1.340 e 1.360 - 1.370 dão a você 10 pontos para o dobro que poderia, caso contrário, fazer menos de 10 pontos no escore.

Após cada jogador ter jogado, os totais acumulativos dos jogadores serão mostrados antes da próxima jogada ser feita (linhas 1.550 - 1.580). Se, entrentanto, qualquer escore exceder o escore do jogo (PT), a linha 1.410 transferirá o comando do programa para a linha 1.540, onde os escores finais são mostrados. Boa sorte!


```
10 REM ************************
20 REM *                      *
30 REM *        JOGO          *
40 REM *                      *
50 REM *   DIREITOS AUTORAIS  *
60 REM *                      *
70 REM *     NELSON ALVES     *
80 REM *                      *
90 REM *                      *
100 REM ************************
110 KEY OFF
120 CLS
130 LOCATE 10,0
140 PRINT "*** JOGO ***"
150 PRINT
160 INPUT "DIGITE UM NUMERO ";X
170 DIM P(10)
180 DIM P$(10)
190 D1 = INT(RND(1)*6)+1
200 D2 = INT(RND(X)*6)+1
210 CLS
220 INPUT "NO. DE JOGADORES (1 A 10) ";NJ
230 IF NJ < 1 THEN 220
240 IF NJ > 10 THEN 220
250 PRINT:INPUT "ESCORE ";PJ
260 IF PJ < 1 THEN 250
270 REM *******************************
280 CLS
290 PRINT "ESTE JOGO PERMITE A CADA"
300 PRINT "JOGADOR MISTURAR OS DADOS"
310 PRINT "E ENTAO TENTAR O DOBRO -"
320 PRINT "SE ELE TIRAR O NO. MAIOR"
330 PRINT "PERDE, ELE TEM UMA CHANCE"
340 PRINT "EM DUAS DE TIRAR O  DOBRO"
350 PRINT "- MAS SE ELE TIRAR  MENOS"
360 PRINT "PERDE, EXISTE 1 CHANCE EM"
370 PRINT "3 DE TIRAR O DOBRO!!!"
380 REM *******************************
390 PRINT
400 PRINT "ESCORE:"
410 PRINT "SEM O DOBRO - TOTAL DOS DADOS "
420 PRINT "DOBRO - DUAS VEZES O TOTAL DOS DADOS"
430 PRINT "FALHOU O DOBRO - NENHUM ESCORE"
440 REM *******************************
```

```
450 PRINT:PRINT:INPUT "PRESSIONE RETURN PARA CONTINUAR ";X$
460 FOR I = 1 TO NJ
470 CLS
480 PRINT "JOGADOR NO.";I;"NOME:";
490 INPUT "";P$(I)
500 P(I) = 0
510 NEXT I
520 PART = 0
530 REM FLAGD = 0 SIGNIFICA NENHUMA TENTATIVA DE DOBRO
540 REM FLAGD = 1 SIGNIFICA TENTATIVA COM O DADO INFERIOR
550 REM FLAGD = 2 SIGNIFICA TENTATIVA COM O DADO SUPERIOR
560 PART = PART + 1
570 FOR I = 1 TO NJ
580 CLS
590 PRINT "************************************"
600 PRINT "PARTIDA NO.";PART
610 PRINT "JOGADOR NO.";I;
620 PRINT SPC(1);"(";P$(I);")"
630 PRINT
640 D1 = INT(6*RND(X))+1
650 D2 = INT(6*RND(X))+1
660 IF D1 <> 0 THEN 680
670 D1 = 1
680 IF D2 <> 0 THEN 700
690 D2 = 1
700 FLAGD = 0
710 PRINT "DADO1 =";D1;SPC(3);"DADO2 =";D2
720 PRINT
730 INPUT "JOGAR NOVAMENTE (N/1/2) ";N12$
740 IF N12$ = "N" THEN 1260
750 IF N12$ = "1" THEN 780
760 IF N12$ = "2" THEN 1030
770 GOTO 730
780 IF D1 <= D2 THEN 810
790 FLAGD = 2
800 GOTO 930
810 FLAGD = 1
820 REM ******************************
830 D1 = INT(RND(X)*3)+1
840 IF D1 = 1 THEN 900
850 IF D2 = 1 THEN 880
860 D1 = D2 - 1
870 GOTO 890
880 D1 = D2 + 1
890 GOTO 1260
```



```
900 D1 = D2
910 GOTO 1260
920 REM ******************************
930 D1 = INT(RND(X)*2)+1
940 IF D1 = 1 THEN 1000
950 IF D2 = 1 THEN 980
960 D1 = D2 - 1
970 GOTO 990
980 D1 = D2 + 1
990 GOTO 1260
1000 D1 = D2
1010 GOTO 1260
1020 REM *****************************
1030 IF D1 <= D2 THEN 1060
1040 FLAGD = 2
1050 GOTO 1180
1060 FLAGD = 1
1070 REM *****************************
1080 D2 = INT(RND(X)*3)+1
1090 IF D2 = 1 THEN 1150
1100 IF D1 = 1 THEN 1130
1110 D2 = D1 - 1
1120 GOTO 1140
1130 D2 = D1 + 1
1140 GOTO 1260
1150 D2 = D1
1160 GOTO 1260
1170 REM *****************************
1180 D2 = INT(RND(X)*2)+1
1190 IF D2 = 1 THEN 1250
1200 IF D1 = 1 THEN 1230
1210 D2 = D1 - 1
1220 GOTO 1240
1230 D2 = D1 + 1
1240 GOTO 1260
1250 D2 = D1
1260 IF FLAGD = 1 THEN 1350
1270 IF FLAGD = 2 THEN 1350
1280 IF D1 <> D2 THEN 1330
1290 TTL = 2*(D1+D2)
1300 IF TTL > 9 THEN 1320
1310 TTL = 10
1320 GOTO 1410
1330 TTL = D1 + D2
1340 GOTO 1410
```

```
1350 IF D1 <> D2 THEN 1400
1360 TTL = 2*(D1+D2)
1370 IF TTL > 9 THEN 1390
1380 TTL = 10
1390 GOTO 1410
1400 TTL = 0
1410 P(I) = P(I) + TTL
1420 CLS
1430 PRINT "DADO1 =";D1;SPC(3);"DADO2 =";D2
1440 PRINT "****************************************"
1450 IF P(I) >= PJ THEN 1540
1460 PRINT "**** TOTAIS ****"
1470 FOR J = 1 TO NJ
1480 PRINT "JOGADOR NO. ";J;SPC(1);P$(J);TAB(23);P(J)
1490 NEXT J
1500 PRINT
1510 INPUT "PRESSIONE RETURN PARA CONTINUAR ";X$
1520 NEXT I
1530 GOTO 560
1540 CLS
1550 PRINT "**** ESCORE FINAL ****"
1560 FOR J = 1 TO NJ
1570 PRINT "JOGADOR NO. ";J;SPC(1);P$(J);TAB(23);P(J)
1580 NEXT J
1590 PRINT
1600 INPUT "JOGAR NOVAMENTE (S/N) ";SN$
1610 IF SN$ = "S" THEN 210
1620 CLS: END
```


JOGO DOS NÚMEROS

O programa **Jogo dos Números (CHUTE)** permite que até 10 jogadores participem dele. O comando **DIM** nas linhas 120 - 150 define cinco matrizes. A variável NUM representa o número randômico que cada jogador deve digitar em cada partida. As variáveis **CHUTE** e **ALBA$** são matrizes bi-dimensionais que mantêm o registro de todos os números digitados por cada jogador e determinam se (ou não) cada número é maior que, ou menor que <> ou <> o número que o jogador colocou. A variável **SC** mantém o registro do escore atual de cada jogador. Cada tentativa soma 1 ao escore; o menor escore ganha. Em cada partida, a variável **GOTIT$** diz ao programa se qualquer jogador ainda não digitou algum número. Note que no início da próxima partida, a matriz GOTIT$ é preenchida com espaços para todos os jogadores nas linhas 450 - 480 — que, incidentalmente, é também onde o programa determina o número randômico NUM para cada jogador na partida.

Em uma tela é gerada a descrição do programa (linhas 180 - 260). As linhas 270 - 380 ajustam o início das variáveis, e as linhas 90 - 310 obtêm o número de jogadores (entre 1 e 10). O maior número disponível que um jogador pode escolher (100, 200, 1.000, ou qualquer que seja a escolha) é determinado nas linhas 320 e 330. Após os escores de todos os jogadores terem sidos colocados em zero, as linhas 370 e 380 selecionam o número máximo de tentativas ("chutes") e o número de partidas (ou números que os jogadoes podem chutar).

O corpo do programa consiste em três *loops* **FOR-NEXT** nas linhas 390 - 860. A variável i controla as partidas; J determina se cada jogador está "chutando" pela primeira vez, segunda etc.; e K controla qual jogador está jogando. Quando cada partida começa, o programa coloca a variável NUMGOTIT em zero e imprime o número de partidas nas linhas 410 - 440. A seguir o *loop* interno de J nas linhas 490 - 850 inicia com 1 e termina com todos os jogadores tendo "chutado" (veja linha 500) ou quando o máximo dado pela variável CHUTES tiver sido alcançado por todos os jogadores (veja linha 490). O *loop* K controla o jogador que está chutando (linhas 550 - 840). O jogo é executado mais ou menos como segue:

Partida 1 inicia
Jogador 1 dá o seu primeiro chute
Jogador 2 dá o seu primeiro chute
Desde que o número de jogadores indicados tenha sido somente dois,
Jogador 1 chuta pela segunda vez
Jogador 2 chuta pela segunda vez
E assim por diante... até ambos os jogadores terem chutado ou atingido o limite
Partida 2 inicia
E assim por diante... até o número de partidas solicitado.

A linha 500 verifica se todos os jogadores chutaram seus números; se positivo, o

controle passa para a linha 860 para a próxima partida. Para todas as partidas a subrotina na linha 880 imprime o escore de todos os jogadores. O número da partida, número chutado, número do jogador e escore são mostrados a cada vez que um jogador chuta um número (linhas 570 - 600). As linhas 620 - 640 dão um sumário dos chutes anteriores dos jogadores, calculando se cada chute foi alto ou baixo <> ou <>, e mostra estes cálculos entre parênteses. O jogador chuta novamente na linha 660. Baseado em se o chute é alto, baixo, ou correto, as linhas 790 - 810, 750 - 770 ou 710 - 730 tomam a decisão apropriada.

No final da partida, o controle retorna para a linha 270, onde você pode começar um novo jogo ou terminar o programa digitando 0 quando for perguntado pelo número de jogadores.

```
10 REM  ***********************
20 REM  *                     *
30 REM  *  JOGO DOS NUMEROS   *
40 REM  *                     *
50 REM  *  DIREITOS AUTORAIS  *
60 REM  *                     *
70 REM  *    NELSON ALVES     *
80 REM  *                     *
90 REM  *                     *
100 REM ***********************
110 KEY OFF
120 DIM NUM(10)
130 DIM CHUTE(10,100)
140 DIM ALBA$(10,100)
150 DIM SC(10)
160 DIM GOTIT$(10)
170 CLS
180 LOCATE 8,5
190 PRINT "*** JOGO DOS NUMEROS ***"
200 PRINT: PRINT: PRINT
210 PRINT "ESTE JOGO PERMITE QUE QUALQUER NUMERO"
220 PRINT "DE JOGADORES DE 1 A 10 CHUTEM UM NO. "
230 PRINT "CADA CHUTE ADICIONA 1 AO ESCORE."
240 PRINT "ASSIM, O MENOR ESCORE GANHARA."
250 PRINT
260 INPUT "PRESSIONE RETURN PARA CONTINUAR ";X$
270 CLS
280 LOCATE 0,5
290 INPUT "NUMERO DE JOGADORES (0 = FIM) ";NJ
300 IF NJ > 10 THEN 270
```



```
310 IF NJ < 1 THEN 950
320 PRINT:INPUT "CHUTE UM NO. DE 1 A ";CHU
330 IF CHU < 1 THEN 950
340 FOR I =1 TO NJ
350 SC(I) = 0
360 NEXT I
370 PRINT: INPUT "NO. MAXIMO DE TENTATIVAS ";CHUTES
380 PRINT: INPUT "NO. DE PARTIDAS ";PARTS
390 FOR I = 1 TO PARTSS
400 RD = I
410 NUMGOTIT = 0
420 CLS: LOCATE 0,5
430 PRINT "PARTIDA NO.";RD
440 PRINT: INPUT "PRESSIONE RETURN PARA CONTINUAR ";X$
450 FOR J =1 TO NJ
460 GOTIT$(J) = CHR$(32)
470 NUM(J) = INT(CHU*RND(1))+1
480 NEXT J
490 FOR J = 1 TO CHUTES
500 IF NUMGOTIT = NJ THEN 860
510 CLS : LOCATE 0,5
520 GOSUB 880
530 FOR K = 1 TO NJ
540 IF GOTIT$(K) = "S" THEN 840
550 CLS
560 LOCATE 0,5
570 PRINT "JOGADOR NO.";K;"(PARTIDA";RD;",TENTATIVA";J;")"
580 PRINT
590 PRINT "ESCORE";SC(K)
600 PRINT
610 IF J = 1 THEN 650
620 FOR L = 1 TO J-1
630 PRINT CHUTE(K,L);"(";ALBA$(K,L);")";
640 NEXT L
650 PRINT
660 PRINT: INPUT "QUAL A SUA TENTATIVA ";G
670 CHUTE(K,J) = G
680 IF G = NUM(K) THEN 710
690 IF G < NUM(K) THEN 750
700 IF G > NUM(K) THEN 790
710 PRINT: PRINT "CORRETO !";CHR$(7);"ELE E";G
720 GOTIT$(K) = "S"
730 NUMGOTIT = NUMGOTIT + 1
740 GOTO 820
750 PRINT: PRINT "ERRADO,";G;"E < QUE O SEU NUMERO"
```

```
760 SC(K) = SC(K) + 1
770 ALBA$(K,J) = "<"
780 GOTO 820
790 PRINT: PRINT "ERRADO,";G;"E > QUE O SEU NUMERO"
800 SC(K) = SC(K) + 1
810 ALBA$(K,J) = ">"
820 PRINT
830 INPUT "PRESSIONE RETURN PARA CONTINUAR ";X$
840 NEXT K
850 NEXT J
860 NEXT I
870 GOTO 270
880 PRINT
890 FOR P = 1 TO NJ
900 PRINT "JOGADOR NO.";P;"SEU ESCORE E";SC(P)
910 NEXT P
920 PRINT
930 INPUT "PRESSIONE ENTER PARA CONTINUAR ";X$
940 RETURN
950 CLS
960 END
```


ROLETA RUSSA

O programa **Roleta Russa (ROLETA)** permite que até 20 jogadores (ou vítimas) joguem. O objetivo, como você já deve saber, é ser o último a ter a cabeça estourada por uma bala.

As linhas 120 - 130 mantêm em uma matriz o registro das vítimas vivas ou mortas e limpa a tela. A seguir o comando **GOSUB** 490 desenha os seis cilindros da arma. Um "0" (zero) representa a bala no sexto cilindro, enquanto o "O" (a letra O) representa os cilindros vazios.

As linhas 170 - 190 acertam o número de jogadores. Se você responder com 0, o programa pula para a linha 500 e termina; caso contrário, a linha 200 coloca em 0 o valor de todos os jogadores na matriz PL, indicando que eles ainda estão vivos. O número de partidas é colocado em zero pela linha 140 e incrementado de 1 na linha 210. As linhas 220 - 480 então permitem que cada jogador rode o cilindro da arma qualquer número de vezes antes de atirar. As linhas 230 - 250 verificam se todos já estão mortos, neste caso as linhas 260 - 280 transferem o controle de volta para a linha 130 para outro jogo.

Se um jogador já foi morto, a linha 290 salta para o próximo jogador. As linhas 300 - 350 então limpam a tela; redesenham o cilindro da arma; mostram o número do jogo e jogador na linha 13, coluna 0; e pergunta quantas vezes o jogador quer rodar o cilindro. O número de voltas que o jogador escolhe determina o número de vezes que o programa randomicamente selecionará os números de (1 a 6). O último número randômico selecionado representa o cilindro que será disparado. Se o cilindro for o primeiro, segundo, terceiro, quarto ou quinto, a arma disparará um CLICK... (SORTUDO); mas se o sexto cilindro for selecionado, a arma disparará a bala e explodirá (linhas 400 - 420). A linha 430 então marcará o jogador atual como morto.

O jogo continua até que um jogador somente permaneça vivo ou todos estejam mortos. Pressione **CTRL C** para terminar o programa, como todos os outros, a qualquer hora.

```
10 REM ***********************
20 REM *                     *
30 REM *    ROLETA  RUSSA    *
40 REM *                     *
50 REM *  DIREITOS AUTORAIS  *
60 REM *                     *
70 REM *    NELSON ALVES     *
80 REM *                     *
90 REM *                     *
100 REM ***********************
```

```
110 KEY OFF
120 DIM PL(20)
130 CLS
140 PART = 0
150 GOSUB 510
160 LOCATE 0,11
170 INPUT "NO. DE JOGADORES (VITIMAS) ";N1
180 IF N1 = 0 THEN 500
190 IF N1 > 20 THEN 130
200 FOR I = 1 TO N1: PL(I) = 0: NEXT I
210 PART = PART + 1
220 FOR I = 1 TO N1
230 FOR J = 1 TO N1
240 IF PL(J) = 0 THEN 290
250 NEXT J
260 PRINT "TODOS ESTAO MORTOS !!!"
270 PRINT: INPUT "PRESSIONE RETURN PARA CONTINUAR ";X$
280 GOTO 130
290 IF PL(I) = 1 THEN 480
300 CLS
310 GOSUB 510
320 LOCATE 0,13
330 PRINT "PARTIDA ";PART
340 PRINT: PRINT "JOGADOR NO.";I
350 PRINT: INPUT "NO. DE VOLTAS DO TAMBOR ";S
360 FOR K = 1 TO S+1
370 X = INT(6*RND(1))+1
380 NEXT K
390 PRINT: PRINT "TAMBOR NO.";X
400 IF X <> 6 THEN 460
410 PRINT: PRINT "BANG ! ! !";CHR$(7)
420 FOR L = 1 TO 4: PRINT CHR$(7);: NEXT L
430 PL(I) = 1
440 PRINT: INPUT "PRESSIONE RETURN PARA CONTINUAR ";X$
450 GOTO 480
460 PRINT: PRINT "CLICK... (SORTUDO)"
470 PRINT: INPUT "PRESSIONE RETURN PARA CONTINUAR ";X$
480 NEXT I
490 GOTO 210
500 CLS: END
510 CLS
520 LOCATE 17,3:PRINT "O"
530 LOCATE 19,4:PRINT "O"
540 LOCATE 19,6:PRINT "O"
550 LOCATE 17,7:PRINT "O"
```



```
560 LOCATE 15,6:PRINT "O"
570 LOCATE 15,4:PRINT "O"
580 LOCATE 0,9:PRINT "CILINDRO 6 (E O) TEM UMA BALA"
590 LOCATE 5,0: PRINT "R O L E T A   R U S S A"
600 RETURN
```

MENSAGEM DE TELA

O programa **Mensagem de Tela (TELA)** permite que você escreva mensagens na tela de forma muito fácil. Duas variações estão disponíveis; para repetir uma linha qualquer número de vezes ou repetir cinco linhas também qualquer número de vezes. (Veja a linha 140). Por exemplo, se você tiver que sair de casa às 16:45, e quiser deixar uma mensagem para uma pessoa de sua família que chegará às 17:00, pode usar o programa TELA para criar uma mensagem que preenche e limpa a tela periodicamente.

Se você selecionar a opção uma linha, a tela será preenchida com aquela linha, um caracter por vez e então apagada a linha um caracter por vez também, da direita para a esquerda e debaixo para cima.

As linhas 180 - 390 geram a variação de uma linha; as linhas 430 - 980, a versão de cinco linhas. O comando **PRINT MID$** move os caracteres, um por vez, da série de caracteres que você digitou, como sendo a mensagem a ser mostrada.

```
10 REM ***********************
20 REM *                     *
30 REM *  MENSAGEM DE TELA   *
40 REM *                     *
50 REM *  DIREITOS AUTORAIS  *
60 REM *                     *
70 REM *    NELSON ALVES     *
80 REM *                     *
90 REM *                     *
100 REM ***********************
110 KEY OFF
120 CLS
130 PRINT
140 INPUT "REPETE 1 LINHA OU 5 (1/5) ";R$
150 IF R$ = "1" THEN 180
160 IF R$ = "5" THEN 430
170 GOTO 140
180 PRINT:INPUT "MSG ";A$
190 X = LEN(A$)
200 IF X > 38 THEN 180
210 PRINT:INPUT "NO. DE LINHAS ";REPS
220 PRINT: INPUT "NO. DE TELAS ";S
230 FOR K = 1 TO S
240 CLS
250 FOR J = 1 TO REPS
260 FOR I = 1 TO X
```



```
270 LIN = J: COL = I
280 LOCATE COL,LIN
290 PRINT MID$(A$,I,1)
300 NEXT I
310 NEXT J
320 FOR J = REPS TO 1 STEP -1
330 FOR I = X TO 1 STEP -1
340 LIN = J:COL = I
350 LOCATE COL,LIN
360 PRINT CHR$(32);
370 NEXT I
380 NEXT J
390 NEXT K
400 PRINT:INPUT "MAIS (S/N) ";SN$
410 IF SN$ = "S" THEN 120
420 GOTO 1000
430 PRINT:INPUT "MSGA ";A$
440 XA = LEN(A$)
450 IF XA > 38 THEN 430
460 PRINT:INPUT "MSGB ";B$
470 XB = LEN(B$)
480 IF XB > 38 THEN 460
490 PRINT:INPUT "MSGC ";C$
500 XC = LEN(C$)
510 IF XC > 38 THEN 490
520 PRINT:INPUT "MSGD ";D$
530 XD = LEN(D$)
540 IF XD > 38 THEN 520
550 PRINT:INPUT "MSGE ";E$
560 XE = LEN(E$)
570 IF XE > 38 THEN 550
580 PRINT: INPUT "CONTINUAR (S/N) ";SN2$
590 IF SN2$ <> "S" THEN 120
600 PRINT:INPUT "ATRASO (0-2000) ";DL
610 PRINT: INPUT "NO. DE REPETICOES ";R2
620 FOR K = 1 TO R2
630 CLS
640 IF XA < 1 THEN 690
650 FOR I = 1 TO XA
660 LIN = 2: COL = I
670 LOCATE COL,LIN: PRINT MID$(A$,I,1);
680 NEXT I
690 IF XB < 1 THEN 740
700 FOR I = 1 TO XB
710 LIN = 3: COL = I
```

```
720 LOCATE COL,LIN: PRINT MID$(B$,I,1);
730 NEXT I
740 IF XC < 1 THEN 790
750 FOR I = 1 TO XC
760 LIN = 4: COL = I
770 LOCATE COL,LIN: PRINT MID$(C$,I,1);
780 NEXT I
790 IF XD < 1 THEN 840
800 FOR I = 1 TO XD
810 LIN = 5: COL = I
820 LOCATE COL,LIN: PRINT MID$(D$,I,1);
830 NEXT I
840 IF XE < 1 THEN 890
850 FOR I =1 TO XE
860 LIN = 6: COL = I
870 LOCATE COL,LIN: PRINT MID$(E$,I,1);
880 NEXT I
890 FOR I = 7 TO 10
900 FOR J = 1 TO 38
910 LIN = I: COL = J
920 LOCATE COL,LIN: PRINT "*";
930 NEXT J
940 NEXT I
950 PRINT
960 IF DL < 1 THEN 980
970 FOR I = 1 TO DL: NEXT I
980 NEXT K
990 GOTO 400
1000 CLS: END
```

Capítulo 3

PROGRAMAS EDUCACIONAIS E PESSOAIS

MÁQUINA DE CALCULAR

O programa **Máquina de Calcular (MÁQUINA)** simula uma máquina de calcular, mas só que ele é muito melhor. Quando você digita os números a serem somados ou subtraídos, são armazenados na memória na mesma ordem na qual você os digitou. Você pode revê-los — junto com o total que está acumulado até este ponto — a qualquer hora e mudar ou eliminar qualquer entrada errônea. Deste modo, a memória do computador tem o mesmo propósito que o papel de uma máquina de somar, entretanto, você tem que subtrair e depois somar de novo para arrumar um valor digitado erroneamente. Com o programa MÁQUINA, você pode ter uma nova lista de números que foram digitados tão logo você tenha feito as correções.

Este programa não precisa de uma impressora, já que até 40 números podem ser facilmente mostrados na tela. Quando todas as entradas são chamadas, elas são mostradas em duas colunas. Por exemplo, se 20 números foram digitados, 10 são mostrados em cada lado da tela, com os números rotulados de 1 a 10 na metade esquerda da tela e os rotulados de 11 a 20 na metade direita. Para corrigir o ítem 13, responda S para a pergunta "IMPRIMIR DETALHES". A seguir você responde 13 ao *prompt* que pergunta pelo número que você quer corrigir e digita a adição ou subtração desejada para repor a entrada incorreta.

As linhas 200 - 340 indicam os comandos que você pode usar. Digitando INSTRUÇÕES as instruções são mostradas novamente. As instruções são exibidas automaticamente na tela no início do programa.

A linha 190 usa o comando **CLS** para limpar a tela.

Toda vez que o cursor é colocado na segunda posição sem um *prompt* anterior, você pode fazer uma nova entrada digitando um número com ou sem o ponto decimal e finalizar número com um + ou -, ou um A ou S. Um A (para adição) e um S (para subtração) podem ser usados para facilitar a entrada de uma adição ou subtração, isto é, as letras são mais fáceis de serem achadas pelo usuário. Com este método, você pode digitar um número após o outro até visualizar um subtotal (S) ou um total (T). A diferença entre os dois é que o T automaticamente limpa a memória e os totais acumulados, enquanto S mostra o total até aquele ponto, permitindo que você continue inserindo números após ver o subtotal, tão bem quanto fazer as correções necessárias.

Os comandos LIGAR e DESLIGAR (impressora) servem para você utilizar ou não a impessora ao invés da tela. Você pode ligar ou desligar a impressora em qualquer ponto do programa. Como mencionado anteriormente, entretanto, você pode usar este programa sem ter uma impressora e desta forma economizar papel.

Outra característica deste programa permite que você tenha resultados intermediários que não interferem com o que está armazenado na memória. Pode-se fazer isto digitando-se um número seguido por uma / (para divisão) ou um X (para multiplicação) e depois digitando-se **RETURN.** A seguir, digitando o número a ser multiplicado ou dividido, seguido por um sinal de igual, e digi-

tando **RETURN**, você terá a resposta. Por exemplo, se você primeiro digitar 10X e um **RETURN**, então 10= e um **RETURN** terão a resposta 10. Você pode somar ou subtrair 100 digitando 100+ ou 100- e digitando **RETURN**.

Você pode adicionar ou subtrair o mesmo número repetitivamente sem redigitá-lo. Apenas insira-o uma vez para ser somado ou subtraido e digite **RETURN** para cada cálculo. Por exemplo, digitando 20- e a seguir **RETURN RETURN RETURN RETURN RETURN RETURN**, você está subtraindo 20 seis vezes. As linhs 520 - 570 ligam a impressora (se um for usado) ou desligam quando você digitar LIGAR ou DESLIGAR. Quando a impressora for ligada, um sinalizador será colocado em S (YN$ na linha 510) ou N (linha 550). O programa sempre será alertado para exibir os dados na impressora ou na tela.

As linhas 620 - 730 avaliam o operador aritmético digitado como o último caracter de um número. Se o caracter não for um A, /, X=, S, + ou -, o operador será considerado inválido e uma mensagem de erro será mostrada (linha 730).

As linhas 750 - 780 avaliam o número digitado antes de um X ou uma / de modo que as linhas 790 e 820 ou 840 provoquem exibição imediata do resultado da divisão na tela e/ou impressora. Do mesmo modo, um produto intermediário de uma multiplicação é impresso nas linhas 860 e 880.

As linhas 900 - 980 acumulam o número digitado em um total, tanto adição como subtração, dependendo do operador final (+, -, A, ou S). A entrada é salva na matriz de memória E$ (linha 1.000). Note que se umS ou um A foi digitado, um - ou um + substituirá o operador digitado na memória.

O subtotal ou total é impresso nas linhas 1.030 - 1.690. O número de entradas contidas na memória (CTR1 na linha 1.230) é dividido por dois para determinar quantas linhas de duas colunas devem ser mostradas ou impressas.

Como previamente mencionado, cada ítem é claramente numerado de modo que qualquer entrada possa ser facilmente localizada e modificada. As linhas 1.700 - 2.020 permitem que você efetue mudanças ou apague as entradas. Para substituir uma entrada por uma correção, você precisa somente digitar um novo número após responder à questão de modificação do detalhe na linha 1.720. A entrada antiga é então ou subtraida do total (se foi uma adição) ou adicionada ao total (se foi uma subtração). O número digitado em substituição é armazenado na mesma localização na matriz de memória.

As linhas 2.030 - 2.050 limpam o acumulador TOTAL e colocam o ponteiro da matriz de memória de volta em zero.

```
10 REM  ********************
20 REM * MAQUINA DE CALCULAR *
30 REM *                    *
40 REM * DIREITOS AUTORAIS  *
50 REM *                    *
60 REM *    NELSON ALVES    *
70 REM *                    *
80 REM **********************
90 REM *                    *
100 REM * SOMA ,  SUBTRAI  E *
110 REM * MANTEM REGISTRO DE *
120 REM *     AUDITORIA      *
130 REM *                    *
140 REM **********************
150 REM
160 DIM E$(200)
170 CTR1 = 0
180 TTL = 0
190 CLS
200 PRINT"*************************************"
210 PRINT"*                                   *"
220 PRINT"*    INSTRUCOES PARA UTILIZACAO     *"
230 PRINT"*                                   *"
240 PRINT"* #/ OU #X DIVISAO OU MULTIPLICACAO *"
250 PRINT"* #= PARA OBTER O RESULTADO         *"
260 PRINT"* S   PARA OBTER O SUBTOTAL         *"
270 PRINT"* T   P/ OBTER TOTAL (LIMPA MEMORIA) *"
280 PRINT"* PARA SOMAR/SUBTRAIR USE +,-,A,S   *"
290 PRINT"* LIGAR         LIGA A IMPRESSORA   *"
300 PRINT"* DESLIGAR      DESLIGA A IMPRESSORA *"
310 PRINT"* FIM           TERMINA O PROGRAMA  *"
320 PRINT"*                                   *"
330 PRINT"*                                   *"
340 PRINT"*************************************"
350 PRINT
360 PRINT
370 PRINT
380 PRINT
390 PRINT"  PRESSIONE A LETRA C PARA CONTINUAR  "
400 IF INKEY$ = "C" THEN GOTO 410 ELSE GOTO 400
410 CLS
420 INPUT"";A$
```



```
430 IF A$ = "INSTRUCOES" THEN 180
440 IF A$ = "FIM" THEN 2060
450 IF A$ = "S" THEN 1020
460 IF A$ = "T" THEN 1020
470 IF A$ = "LIGAR" THEN 500
480 IF A$ = "DESLIGAR" THEN 540
490 GOTO 580
500 IF SN$ = "S" THEN 530
510 SN$ = "S"
520 REM ***** IMPRESSORA LIGADA *****
530 GOTO 420
540 IF SN$ = "N" THEN 570
550 SN$ = "N"
560 REM **** IMPRESSORA DESLIGADA ****
570 GOTO 420
580 LA = LEN(A$)
590 IF LA > O THEN 620
600 A$ = E$(CTR1)
610 LA = LEN(A$)
620 OPER$ = MID$(A$,LA,1)
630 IF OPER$ = "A" THEN 690
640 IF OPER$ = "/" THEN 750
650 IF OPER$ = "X" THEN 750
660 IF OPER$ = "=" THEN 790
670 IF OPER$ = "S" THEN 710
680 GOTO 720
690 OPER$ = "+"
700 GOTO 720
710 OPER$ = "-"
720 IF OPER$ = "+" OR OPER$ = "-" THEN 900
730 PRINT"ERRO - ULTIMO CARACTERE DEVE SER A,S,+,-"
740 GOTO 420
750 NUM$ = MID$(A$,1,LA-1)
760 NA = VAL(NUM$)
770 OA$ = OPER$
780 GOTO 420
790 NUM$ = MID$(A$,1,LA-1)
800 NB = VAL(NUM$)
810 IF OA$ = "X" THEN 860
820 PRINT NA/NB
830 IF SN$ <> "S" THEN 850
840 LPRINT NA/NB
850 GOTO 420
860 PRINT NA*NB
870 IF SN$ <> "S" THEN 420
```

```
880 LPRINT NA*NB
890 GOTO 420
900 NUM$ = MID$(A$,1,LA-1)
910 N1 = VAL(NUM$)
920 N1 = (INT(N1*100+.5))/100
930 IF OPER$ = "+" THEN 960
940 TTL = TTL-N1
950 GOTO 970
960 TTL = TTL+N1
970 CTR1 = CTR1+1
980 TTL = (INT(TTL*100+.5))/100
990 NX$ = STR$(N1)
1000 E$(CTR1) = NX$+OPER$
1010 GOTO 420
1020 IF SN$ <> "S" THEN 1100
1030 IF A$ = "S" THEN 1070
1040 LPRINT "TOTAL = ";
1050 LPRINT TTL
1060 GOTO 1160
1070 LPRINT "SUBTOTAL = ";
1080 LPRINT TTL
1090 GOTO 1160
1100 IF A$ = "S" THEN 1140
1110 PRINT "TOTAL = ";
1120 PRINT TTL
1130 GOTO 1160
1140 PRINT "SUBTOTAL = ";
1150 PRINT TTL
1160 INPUT "IMPRIMIR DETALHES ";ISN$
1170 IF A$ = "S" THEN 1190
1180 IF A$ = "T" THEN 1210
1190 IF ISN$ <> "S" THEN 420
1200 GOTO 1220
1210 IF ISN$ <> "S" THEN 2030
1220 C2 = 0
1230 CA = CTR1/2
1240 IF CA = INT(CA) THEN 1260
1250 CA = INT(CA)+1
1260 FOR I = 1 TO CA
1270 I$ = STR$(I)
1280 LI = LEN(I$)
1290 IF CTR1 = 0 THEN 420
1300 I2$ = I$+"."
1310 IF SN$ <> "S" THEN 1510
1320 LPRINT I2$; SPC(5-LI); E$(I);
```



```
1330 LPRINT SPC(15-LEN(E$(I)));
1340 IX = I+CA
1350 IF IX > CTR1 THEN 1400
1360 IX$ = STR$(IX)
1370 LX = LEN(IX$)
1380 I2$ = IX$+"."
1390 LPRINT I2$; SPC(5-LX); E$(IX);
1400 LPRINT
1410 NEXT I
1420 LPRINT
1430 IF A$ = "S" THEN 1470
1440 LPRINT "TOTAL = ";
1450 LPRINT TTL
1460 GOTO 1490
1470 LPRINT "SUBTOTAL = ";
1480 LPRINT TTL
1490 LPRINT"*************************************"
1500 GOTO 1690
1510 PRINT I2$; SPC(5-LI); E$(I);
1520 PRINT SPC(15-LEN(E$(I)));
1530 IX = I+CA
1540 IF IX > CTR1 THEN 1590
1550 IX$ = STR$(IX)
1560 LX = LEN(IX$)
1570 I2$ = IX$+"."
1580 PRINT I2$; SPC(5-LX); E$(IX);
1590 PRINT
1600 NEXT I
1610 PRINT
1620 IF A$ = "S" THEN 1660
1630 PRINT "TOTAL = ";
1640 PRINT TTL
1650 GOTO 1680
1660 PRINT "SUBTOTAL = ";
1670 PRINT TTL
1680 PRINT"*************************************"
1690 IF A$ = "T" THEN 2030
1700 INPUT "MUDA DETALHE ";DSN$
1710 IF DSN$ <> "S" THEN 420
1720 INPUT "DETALHE #: ";DN
1730 IF DN > CTR1 THEN 1700
1740 INPUT "NOVO VALOR: ";A$
1750 LA = LEN(A$)
1760 IF LA = 0 THEN 1740
1770 OPER$ = MID$(A$,LA,1)
```

```
1780 IF OPER$ = "A" THEN 1810
1790 IF OPER$ = "S" THEN 1830
1800 GOTO 1840
1810 OPER$ = "+"
1820 GOTO 1840
1830 OPER$ = "-"
1840 IF OPER$ = "+" OR OPER$ = "-" THEN 1870
1850 PRINT"ERRO - ULTIMO CARACTERE DEVE SER A,B,+,-"
1860 GOTO 1740
1870 NUM$ = MID$(A$,1,LA-1)
1880 N1 = VAL(NUM$)
1890 LE = LEN(E$(DN))
1900 OE$ = MID$(E$(DN),LE,1)
1910 NE$ = MID$(E$(DN),1,LE-1)
1920 NE = VAL(NE$)
1930 IF OE$ = "+" THEN 1960
1940 TTL = TTL+NE
1950 GOTO 1970
1960 TTL = TTL-NE
1970 IF OPER$ = "+" THEN 2000
1980 TTL = TTL-N1
1990 GOTO 2010
2000 TTL = TTL+N1
2010 E$(DN) = NUM$+OPER$
2020 GOTO 1700
2030 CTR1 = 0
2040 TTL = 0
2050 GOTO 420
2060 CLS: END
```



```
SUBTOTAL =  127
 1.    12+         5.      25-
 2.    14+         6.      123+
 3.    25-         7.      124-
 4.    152+

SUBTOTAL =  127
************************************
SUBTOTAL =  147
 1.    32+         5.      25-
 2.    14+         6.      123+
 3.    25-         7.      124-
 4     152+

SUBTOTAL =  147
************************************
TOTAL =  147
 1     32+         5.      25-
 2.    14+         6.      123+
 3.    25-         7.      124
 4.    152+

TOTAL =  147
************************************
```

Figura 4. Programa Máquina de Calcular: exemplo de saída do programa.

ADIÇÃO/SUBTRAÇÃO

O programa **Adição/Subtração (ADISUB)** foi desenvolvido para melhorar a *performance* na adição e subtração.

O programa desenha um quadro no meio da tela (se desejado) na linha 150). Respondendo com qualquer número de 2 a 999 nas linhas 160 a 190, o usuário pode facilmente selecionar as faixas superior e inferior a serem somadas ou subtraidas.

Os totais das respostas corretas e incorretas encontram-se nas linhas 220 e 230. Você pode escolher adição somente, subtração somente, ou uma mistura dos dois — selecionando randomicamente.

A rotina das linhas 300 - 430 desenham o quadro em torno do problema (se você digitar S na linha 150). Um pequeno caracter é utilizado CHR$(189) e repetido para desenhar o quadro.

O traço (sublinha) é usado para desenhar a barra abaixo da conta, assim como você faz no papel. O processo de seleção randômica feito para o primeiro e segundo números é feito nas linhas 440 e 450. Se você quer um número randômico entre 1 e 50, por exemplo, use o comando

10 N1 = INT(50*RND(1))+1

Neste programa, o comando **INPUT** da linha 140 dá um valor para a variável X. Qualquer número positivo como gerador causa a seleção do número randômico.

Para evitar respostas negativas, o valor N é dado para NEG$ na linha 120. Para mostrar a conta no meio da tela,os dois números são convertidos para séries de caracteres *(strings)* em N1$ e N2$. As linhas 650 e 660 determinam o comprimento destes dois ítens alfanuméricos (outra palavra para série de caracteres). Os caracteres, primeiro e segundo números, são então colocados dentro, um dígito por vez, da vigésima posição da tela, da direita para a esquerda. Esta movimentação é realizada por dois *loops* **FOR-NEXT** que têm um passo negativo (isto é, decrementam do comprimento da série de caracteres até 1). Ao mesmo tempo, os comandos **LOCATE** nas linhas 700 e 760 causam a movimentação de cada dígito para a vigésima posição da ela, a seguir a décima-nona e assim por diante nas linhas 9 e 8.

As linhas 670 a 830 são similares às linhas 300 - 430 posições em que os dígitos dos números N1 e N2 são determinados pelas linhas 700 e 760. As linhas 790 - 840 desenham o operador + ou - e sublinham a conta.

Após a conta ser mostrada, você será solicitado a digitar uma resposta. Se a resposta que você der for correta, o programa vai para outra conta; se sua resposta estiver incorreta, a resposta correta é apresentada. Você deve pressionar RETURN para continuar.

Para terminbar o programa, digite 99999 como resposta para a conta. Um resumo das suas respostas corretas e incorretas será então mostrado.

```
10 REM ************************
20 REM *                      *
30 REM *  ADICAO/SUBTRACAO    *
40 REM *                      *
50 REM *  DIREITOS AUTORAIS   *
60 REM *                      *
70 REM *     NELSON ALVES     *
80 REM *                      *
90 REM *                      *
100 REM ************************
110 KEY OFF
120 NEG$ = "N"
130 CLS
140 INPUT "NUMERO RANDOMICO (1 A 9999)";X
150 INPUT "COM REQUADRO (S/N)";SN$
160 INPUT "PRIMEIRO NUMERO DE 1 A (2/999)";T1
170 IF T1 < 2 THEN 160
180 IF T1 > 999 THEN 160
190 INPUT "SEGUNDO NUMERO DE 1 A (2/999)";B1
200 IF B1 < B2 THEN 190
210 IF B1 > 999 THEN 190
220 ERS = 0
230 C1 = 0
240 INPUT "ADICIONAR, SUBTRAIR, AMBOS (A,S,D)";ASB$
250 IF ASB$ = "A" THEN 290
260 IF ASB$ = "S" THEN 290
270 IF ASB$ = "D" THEN 290
280 GOTO 240
290 CLS: IF SN$ = "N" THEN 440
300 R = 13
310 FOR C = 5 TO 15
320 LOCATE R,C
330 PRINT CHR$(189);
340 LOCATE (R+10),C
350 PRINT CHR$(189);
360 NEXT C
370 C = 5
380 FOR R = 13 TO 23
390 LOCATE R,C
400 PRINT CHR$(189);
```

```
410 LOCATE R,(C+10)
420 PRINT CHR$(189);
430 NEXT R
440 N1 = INT(T1*RND(X))+1
450 N2 = INT(B1*RND(X))+1
460 IF NEG$ <> "N" THEN 510
470 IF N1 > N2 THEN 510
480 NX = N1
490 N1 = N2
500 N2 = NX
510 IF ASB$ = "S" THEN 570
520 IF ASB$ = "A" THEN 550
530 O1 = INT(2*RND(X))+1
540 GOTO 580
550 O1 = 1
560 GOTO 580
570 O1 = 2
580 IF O1 = 1 THEN 590
590 N1$ = STR$(N1)
600 N2$ = STR$(N2)
610 IF O1 = 1 THEN 640
620 OP$ = "-"
630 GOTO 650
640 OP$ = "+"
650 L1 = LEN(N1$)
660 L2 = LEN(N2$)
670 J = 20
680 FOR I = L1 TO 1 STEP -1
690 J = J-1
700 LOCATE J,8
710 PRINT MID$(N1$,I,1);
720 NEXT I
730 J = 20
740 FOR I = L2 TO 1 STEP -1
750 J = J-1
760 LOCATE J,9
770 PRINT MID$(N2$,I,1)
780 NEXT I
790 LOCATE 16,9
800 PRINT OP$;
810 FOR I = 16 TO 19
820 LOCATE I,10
830 PRINT "-";
840 NEXT I
850 LOCATE 0,23
```



```
860 INPUT "DIGITE (99999 PARA FINALIZAR): ";X1
870 IF X1 = 99999! THEN 1030
880 IF O1 = 2 THEN 910
890 IF X1 = N1+N2 THEN 990
900 GOTO 920
910 IF X1 = N1-N2 THEN 990
920 CLS
930 IF O1 = 2 THEN 960
940 PRINT "ERRADO ";N1; " + ";N2; " = ";N1+N2
950 GOTO 970
960 PRINT "ERRADO ";N1; " - ";N2; " = "; N1-N2
970 ERS = ERS+1
980 GOTO 1010
990 C1 = C1+1
1000 GOTO 290
1010 INPUT X$
1020 GOTO 290
1030 CLS
1040 PRINT "**************************************"
1050 PRINT "*"; TAB(38);"*"
1060 PRINT "*          CORRETO = ";C1;TAB(38); "*"
1070 PRINT "*"; TAB(38); "*"
1080 PRINT "*          ERROS = "; ERS; TAB(38); "*"
1090 PRINT "*"; TAB(38); "*"
1100 IF C1 > 0 THEN 1130
1110 IF ERS > 0 THEN 1130
1120 ERS = 1
1130 PRINT "*          ESCORE = "; INT(C1/(C1+
     ERS)*100); "%"; TAB(38); "*"
1140 PRINT "*"; TAB(38); "*"
1150 PRINT "**************************************"
1160 PRINT
1170 PRINT
1180 PRINT
1190 PRINT
1200 END
```

CALCULADORA

O programa **Calculadora (CALCULA)** funciona como uma calculadora — somando, subtraindo, multiplicando, dividindo, tirando a raiz e elevando vários números. Na inicialização do programa, a linha 130 limpa o acumulador de totais (TTL) para zero. Cada cálculo realizado é digitado na linha 140. O que quer que seja que você digite é então avaliado pelas linhas 160 até o final do programa. Após cada avaliação, o controle é devolvido à linha 140 para a próxima operação.

Você pode entrar com dois números, os quais são separados por um operador e terminados por um sinal de igual, como em

200-100=

Quando um número inicia a expressão digitada, o valor total acumulado anteriormente (TTL) é automaticamente limpo. Isto é determinado nas linhas 280 a 330, onde o primeiro caracter digitado é inspecionado para ver se ele é numérico ou um operador.

As linhas 360 - 490 asseguram que a operação seja válida (ambos X, /, +, -, ^, A, S, um ponto ou um número).

Então, dependendo se o número que está sendo lido caracter a caracter é o primeiro ou o segundo número, as linhas 500 - 630 movem os números e/ou pontos decimais para N1$ ou N2$. A série de caracteres é examinada pelo comprimento da série de caracteres (LA na linha 340). Quando isto for completado, N1 e N2 receberão valores numéricos das séries de caracteres N1$ e N2$ (linhas 710 e 740). As linhas 750 e 760 limpam o total se o primeiro caracter digitado for numérico. As linhas 770 - 960 então computam o total para o resultado de N1 e N2, produzido pelo primeiro operador. Então, se o segundo operador for um sinal de igual, o resultado será impresso (linhas 940 e 950). Entretanto, se o primeiro caracter for um operador, o total anterior será somado, multiplicado, e assim por diante, para obter um novo total. Mais uma vez, sempre que aparecer um sinal de igual, o total será mostrado na tela.

A seguir temos comandos de particular interesse.

A linha 350 avalia os caracteres digitados, um por vez. A instrução MID$(A$,I,1) representa o segmento de A$, começando na posição número I para um comprimento de 1 caracter. Note que i é incrementado de 1 no *loop* **FOR-NEXT** nas linhas 340 - 680.

A função LEN(A$) na linha 150 determina o comprimento de A$ de modo que ele possa ser avaliado cracter por caracter. Finalmente, lembre-se de que A e S podem ser usados no lugar de + e - alternativamente.


```
10 REM  ************************
20 REM  *                      *
30 REM  *     CALCULADORA      *
40 REM  *                      *
50 REM  * DIREITOS AUTORAIS    *
60 REM  *                      *
70 REM  *     NELSON ALVES     *
80 REM  *                      *
90 REM  *                      *
100 REM ************************
110 KEY OFF
120 CLS
130 TTL = 0
140 INPUT A$
150 LA = LEN(A$)
160 IF LA = 0 THEN 140
170 IF A$ = "FIM" THEN 1190
180 IF A$ <> "C" THEN 220
190 TTL = 0
200 CLS
210 GOTO 140
220 N1$ = ""
230 N2$ = ""
240 O1$ = ""
250 O2$ = ""
260 C1N1 = 0
270 C2N2 = 0
280 IF MID$(A$,1,1) >= "0" THEN 300
290 GOTO 310
300 IF MID$(A$,1,1) <= "9" THEN 330
310 FLOPER$ = "S"
320 GOTO 340
330 FLOPER$ = "N"
340 FOR I = 1 TO LA
350 A2$ = MID$(A$,I,1)
360 IF A2$ = "X" THEN 640
370 IF A2$ = "/" THEN 640
380 IF A2$ = "^" THEN 640
390 IF A2$ = "+" THEN 640
400 IF A2$ = "-" THEN 640
410 IF A2$ = "A" THEN 640
420 IF A2$ = "S" THEN 640
430 IF A2$ = "=" THEN 640
440 IF A2$ = "." THEN 500
450 IF A2$ > "9" THEN 480
```

```
460 IF A2$ < "0" THEN 480
470 GOTO 500
480 PRINT "DEVE SER # OU X,/,^,+,-,A,S,="
490 GOTO 140
500 IF O1$ = "" THEN 580
510 IF FLOPER$ = "S" THEN 580
520 C2N2 = C2N2 + 1
530 IF C2N2 > 1 THEN 560
540 N2$ = MID$(A2$,1,1)
550 GOTO 570
560 N2$ = N2$+MID$(A2$,1,1)
570 GOTO 680
580 C1N1 = C1N1+1
590 IF C1N1 > 1 THEN 620
600 N1$ = MID$(A2$,1,1)
610 GOTO 630
620 N1$ = N1$+MID$(A2$,1,1)
630 GOTO 680
640 IF O1$ = "" THEN 670
650 O2$ = A2$
660 GOTO 680
670 O1$ = A2$
680 NEXT I
690 IF LEN(N1$) < 1 THEN 720
700 IF N1$ = "" THEN 720
710 N1 = VAL(N1$)
720 IF LEN(N2$) < 1 THEN 750
730 IF N2$ = "" THEN 750
740 N2 = VAL(N2$)
750 IF FLOPER$ = "S" THEN 970
760 TTL = 0
770 IF O1$ = "A" THEN 850
780 IF O1$ = "+" THEN 850
790 IF O1$ = "S" THEN 870
800 IF O1$ = "-" THEN 870
810 IF O1$ = "X" THEN 890
820 IF O1$ = "/" THEN 910
830 IF O1$ = "^" THEN 930
840 GOTO 140
850 TTL = N1 + N2
860 GOTO 940
870 TTL = N1 - N2
880 GOTO 940
890 TTL = N1 * N2
```



```
900 GOTO 940
910 TTL = N1 / N2
920 GOTO 940
930 TTL = N1 ^ N2
940 IF O2$ <> "=" THEN 140
950 PRINT "* TOTAL *";TTL
960 GOTO 140
970 IF O1$ = "=" THEN 1170
980 IF O1$ = "+" THEN 1060
990 IF O1$ = "A" THEN 1060
1000 IF O1$ = "-" THEN 1080
1010 IF O1$ = "S" THEN 1080
1020 IF O1$ = "X" THEN 1100
1030 IF O1$ = "/" THEN 1120
1040 IF O1$ = "^" THEN 1140
1050 GOTO 140
1060 TTL = TTL + N1
1070 GOTO 1150
1080 TTL = TTL - N1
1090 GOTO 1150
1100 TTL = TTL * N1
1110 GOTO 1150
1120 TTL = TTL / N1
1130 GOTO 1150
1140 TTL = TTL ^ N1
1150 IF O2$ = "=" THEN 1170
1160 GOTO 140
1170 PRINT "* TOTAL *";TTL
1180 GOTO 140
1190 CLS: END
```

CALENDÁRIO

O programa **Calendário (CALEND)** produz um calendário para qualquer ano de 1800 a 2099. Com este programa, você pode achar a data de aniversário de qualquer pessoa, o dia e a semana exatos, ou imprimir o calendário de qualquer ano quantas vezes quiser.

A lógica por detrás deste cálculo é explicada nas linhas 140 - 180. A cada dia da semana é dado um número (Domingo e 1, Segunda e 2, e assim por diante, até Sábado, o qual é 7). Em cada ano sucessivo, 1 de janeiro cai um dia maior do que o ano anterior. As únicas excessões para esta regra são:

1. Dois dias são somados no ano seguinte ao ano bissexto.
2. Nem todos os anos do século são bissextos embora eles sejam divisíveis por quatro.

Estes anos centesimais, os quais terminam com dois zeros, são anos bissextos somente se eles forem divisíveis por 400. Deste modo, 1800 e 1900 não foram anos bissextos, embora o ano 2000 venha a ser ano bissexto.

Como você pode ver, este programa mantém calendários dos anos de 1800 a 2099. Se você quiser estender-se para além de 2099 e abaixo de 1800, as linhas 360 - 450 ajustam os parâmetros para 1800, as linhas 460 - 540 para os anos 1900, e as linhas 550 - 630 para os anos 2000. Para adicionar outros séculos, você precisa somar também módulos para o início e fim dos anos. Note que R1 inicia em 4 na linha 390 indicando que 1 de janeiro em 1800 caiu na quarta-feira. A variável ANO na linha 400 é 1, significando que 1800 não era um ano bissexto. E, na linha 430, LF (sinalizador do ano bissexto) é 99, o que significa que 1800 não foi ano bissexto naquele século.

Desta informação básica, contando de 1800, 1900 ou 2000 para o ano do calendário desejado, você pode facilmente calcular exatamente que o dia foi o dia 1 de janeiro em qualquer ano e se ou não este ano foi um ano bissexto. A geração de um calendário por ano é portanto uma tarefa simples.

Agora vamos discutir alguns comandos específicos do programa. Como mencionamos anteriormente, as linhas 200 - 630 ajustam os parâmetros desejados para o primeiro ano do século para o ano do calendário requisitado. Você notará que, na linha 200, você pode finalizar o programa digitando 0 para o ano desejado. Obviamente este programa, como qualquer outro, pode também ser terminado através do comando **CTRL + C.**

As linhas 640 - 860 contêm um *loop* **FOR-NEXT** que computa o dia da semana para 1 de janeiro do ano pedido. O *loop* inicia (linha 640) com FANO — definido anteriormente como 1800, 1900 ou 2000 — e termina quando o ano sendo computado for igual ao ano requerido (linha 850 quando o contador I = Y1, o ano digitado na linha 200). A variável DIA1 (o dia da semana do primeiro dia do ano) é somada, uma a uma, quando cada ano é contado, exceto quando o ano anterior tiver sido bissexto. Neste caso, 2 é somado a DIA1 (linhas 740 - 750). Se DIA1 exceder 7 o valor é mudado para DIA1 menos 7 (linhas 810 - 820).

Note que na linha 840, o resíduo de um comando de depuração ainda existe com o programa. Neste caso os valores de LF e DIA1 foram mostrados na tela para determinar o que estava acontecendo com a lógica do programa. O comando **PRINT** pode ajudar a mostrar os valores de algumas variáveis chave que você pode estar querendo conhecer.

Após o dia da semana do primeiro dia do ano ter sido determinado e se ou não o ano selecionado for bissexto, o programa estará pronto para gerar o calendário. As linhas 940 - 2.030 contêm dois *loops*, o *loop* I que conta de 1 até 12 para os meses, e o *loop* J que conta de 1 a 31 para os dias de cada mês. As linhas 990 - 1.330 fornece as abreviações dos meses sendo mostrados. As linhas 950 - 960 geram a impressão do calendário. CALEND, como qualquer outro programa, pode ser usado com ou sem uma impressora. Obviamente, o programa fica mais prático com sua própria impressora.

As linhas 1.450 - 1.630 determinam quantos dias há em cada mês. assim como o dia da semana é incrementado de 1 (de 1 a 7) e o número do dia do mês é impresso na coluna do dia da semana espaço pela instrução SPC (linhas 1.680 - 1.970). A linha 1.680 usa o comando **ON GOTO** que gera um salto para a linha desejada quando R1 - 1, 2, 3, 4, 5, 6 ou 7. Quando o calendário está completo, a linha 2.040 fecha a impressora (se ela estiver sendo utilizada) e a linha 2.050 retorna o controle para a linha 200.

```
10 REM ************************
20 REM *                      *
30 REM *      CALENDARIO      *
40 REM *                      *
50 REM * DIREITOS AUTORAIS    *
60 REM *                      *
70 REM *    NELSON ALVES      *
80 REM *                      *
90 REM *                      *
100 REM ************************
110 KEY OFF
130 CLS
140 REM 1980 E BASE PARA LA/3 (TER)
150 REM 1981 E BASE PARA LA+1/5 (QUA)
160 REM 1982 E BASE PARA LA+2/6 (SEX)
170 REM 1983 E BASE PARA LA+3/7 (SAB)
180 REM 01/01/80 CAI NO TERCEIRO DIA DA SEMANA, TERCA
190 CLS
```

```
200 INPUT "ANO DESEJADO PARA O CALENDARIO ";A1
210 IF A1 =0 THEN 2620
220 IF A1 < 1800 THEN 190
230 IF A1 > 2099 THEN 190
240 ANO1 = INT(A1/4)
250 L1 = ANO1*4
260 ANO = A1-L1
270 REM ANO = 0 (ANO BISSEXTO)
280 REM ANO = 1 (LA + 1)
290 REM ANO = 2 (LA + 2)
300 REM ANO = 3 (LA + 3)
310 REM 1800 E 1900 NAO SAO ANOS BISSEXTOS
320 IF A1 < 1900 THEN 360
330 IF A1 < 2000 THEN 460
340 IF A1 < 2100 THEN 550
350 REM ****************************
360 IANO = 1801
370 FANO = 1899
380 IF A1 <> 1800 THEN 420
390 R1 = 4
400 ANO = 1
410 GOTO 870
420 DIA1 = 4
430 LF = 99
440 REM * LA = 99 SIGNIFICA ANO NO MESMO SECULO *
450 GOTO 640
460 IANO = 1901
470 FANO = 1999
480 IF A1 <> 1900 THEN 520
490 R1 = 2
500 ANO = 1
510 GOTO 870
520 DIA1 = 2
530 LF = 99
540 GOTO 640
550 IANO = 2001
560 FANO = 2099
570 IF A1 <> 2000 THEN 610
580 R1 = 7
590 ANO = 0
600 GOTO 870
610 DIA1 = 7
620 LF = 0
630 GOTO 640
640 FOR I = IANO TO FANO
```



```
650 LF = LF+1
660 IF LF > 90 THEN 780
670 IF LF < 4 THEN 740
680 LF = 0
690 REM LF = 0 SIGNIFICA LA
700 REM LF = 1 SIGNIFICA LA + 1
710 REM LF = 2 SIGNIFICA LA + 2
720 REM LF = 3 SIGNIFICA LA + 3
730 REM LF = 99 SIGNIFICA ANO NO MESMO SECULO
740 IF LF <> 1 THEN 780
750 DIA1 = DIA1 + 2
760 GOTO 810
770 DIA1 = 5
780 DIA1 = DIA1 + 1
790 IF LF < 90 THEN 810
800 LF = 1
810 IF DIA1 < 8 THEN 830
820 DIA1 = DIA1 - 7
830 R1 = DIA1
840 REM PRINT "LF = ";LF;"DIA1 = ";DIA1
850 IF I = A1 THEN 870
860 NEXT I
870 REM R1 E O RESTANTE (DIA PARA 1 DE JANEIRO)
880 R1 = R1 - 1
890 IF R1 > 0 THEN 910
900 R1 = 7
910 REM PRINT "R1 = ";R1
920 INPUT "IMPRIMIR (S/N) ";X$
930 CLS
940 FOR I = 1 TO 12
950 IF X$ <> "S" THEN 980
960 LPRINT
970 GOTO 980
980 PRINT
990 MO$ = ""
1000 IF I = 1 THEN 1120
1010 IF I = 2 THEN 1140
1020 IF I = 3 THEN 1160
1030 IF I =4 THEN 1180
1040 IF I = 5 THEN 1200
1050 IF I = 6 THEN 1220
1060 IF I = 7 THEN 1240
1070 IF I = 8 THEN 1260
1080 IF I = 9 THEN 1280
1090 IF I = 10 THEN 1300
```

```
1100 IF I = 11 THEN 1320
1110 IF I = 12 THEN 1340
1120 ME$ = "JANEIRO"
1130 GOTO 1350
1140 ME$ = "FEVEREIRO"
1150 GOTO 1350
1160 ME$ = "MARCO"
1170 GOTO 1350
1180 ME$ = "ABRIL"
1190 GOTO 1350
1200 ME$ = "MAIO"
1210 GOTO 1350
1220 ME$ = "JUNHO"
1230 GOTO 1350
1240 ME$ = "JULHO"
1250 GOTO 1350
1260 ME$ = "AGOSTO"
1270 GOTO 1350
1280 ME$ = "SETEMBRO"
1290 GOTO 1350
1300 ME$ = "OUTUBRO"
1310 GOTO 1350
1320 ME$ = "NOVEMBRO"
1330 GOTO 1350
1340 ME$ = "DEZEMBRO"
1350 IF X$ <> "S" THEN 1450
1360 IF I <> 7 THEN 1400
1370 FOR K = 1 TO 10
1380 LPRINT
1390 NEXT K
1400 LPRINT SPC(1);ME$;"/";A1
1410 LPRINT "-------------------------------":
1420 LPRINT SPC(1);"DO";SPC(2);"SE";SPC(2);"TE";
1430 LPRINT SPC(2);"QA";SPC(2);"QI";SPC(2);"SE";SPC(2);"SA"
1440 GOTO 1490
1450 PRINT SPC(1);ME$;"/";A1
1460 PRINT "-------------------------------":
1470 PRINT SPC(1);"DO";SPC(2);"SE";SPC(2);"TE";
1480 PRINT SPC(2);"QA";SPC(2);"QI";SPC(2);"SE";SPC(2);"SA"
1490 FOR J = 1 TO 31
1500 IF J < 28 THEN 1700
1510 IF I = 1 THEN 1700
1520 IF I = 3 THEN 1670
1530 IF I = 5 THEN 1670
1540 IF I = 7 THEN 1670
```



```
1550 IF I = 8 THEN 1670
1560 IF I = 10 THEN 1670
1570 IF I = 12 THEN 1670
1580 IF I = 9 THEN 1690
1590 IF I = 4 THEN 1690
1600 IF I = 6 THEN 1690
1610 IF I = 11 THEN 1690
1620 IF ANO = 0 THEN 1650
1630 IF J > 28 THEN 2430
1640 GOTO 1700
1650 IF J > 29 THEN 2430
1660 GOTO 1700
1670 IF J > 31 THEN 2430
1680 GOTO 1700
1690 IF J > 30 THEN 2430
1700 R1 = R1+1
1710 IF R1 < 8 THEN 1770
1720 R1 = 1
1730 IF X$ <> "S" THEN 1760
1740 LPRINT
1750 GOTO 1770
1760 PRINT
1770 ON R1 GOTO 1780,1840,1940,2040,2140,2240,2340
1780 IF X$ <> "S" THEN 1810
1790 LPRINT SPC(0);J;
1800 GOTO 1820
1810 PRINT SPC(0);J;
1820 GOSUB 2500
1830 GOTO 2430
1840 IF X$ <> "S" THEN 1890
1850 IF J <> 1 THEN 1870
1860 LPRINT SPC(4);
1870 LPRINT J;
1880 GOTO 1920
1890 IF J <> 1 THEN 1910
1900 PRINT SPC(4);
1910 PRINT J;
1920 GOSUB 2500
1930 GOTO 2430
1940 IF X$ <> "S" THEN 1990
1950 IF J <> 1 THEN 1970
1960 LPRINT SPC(8);
1970 LPRINT J;
1980 GOTO 2020
1990 IF J <> 1 THEN 2010
```

```
2000 PRINT SPC(8);
2010 PRINT J;
2020 GOSUB 2500
2030 GOTO 2430
2040 IF X$ <> "S" THEN 2090
2050 IF J <> 1 THEN 2070
2060 LPRINT SPC(12);
2070 LPRINT J;
2080 GOTO 2120
2090 IF J <> 1 THEN 2110
2100 PRINT SPC(12);
2110 PRINT J;
2120 GOSUB 2500
2130 GOTO 2430
2140 IF X$ <> "S" THEN 2190
2150 IF J <> 1 THEN 2170
2160 LPRINT SPC(16);
2170 LPRINT J;
2180 GOTO 2220
2190 IF J <> 1 THEN 2210
2200 PRINT SPC(16);
2210 PRINT J;
2220 GOSUB 2500
2230 GOTO 2430
2240 IF X$ <> "S" THEN 2290
2250 IF J <> 1 THEN 2270
2260 LPRINT SPC(20);
2270 LPRINT J;
2280 GOTO 2320
2290 IF J <> 1 THEN 2310
2300 PRINT SPC(20);
2310 PRINT J;
2320 GOSUB 2500
2330 GOTO 2430
2340 IF X$ <> "S" THEN 2390
2350 IF J <> 1 THEN 2370
2360 LPRINT SPC(24);
2370 LPRINT J;
2380 GOTO 2420
2390 IF J <> 1 THEN 2410
2400 PRINT SPC(24);
2410 PRINT J;
2420 GOSUB 2500
2430 NEXT J
2440 IF X$ <> "S" THEN 2470
```



```
2450 LPRINT
2460 GOTO 2480
2470 PRINT
2480 NEXT I
2490 GOTO 190
2500 J$ = STR$(J)
2510 LJ = LEN(J$)
2520 IF X$ <> "S" THEN 2570
2530 IF LJ < 3 THEN 2560
2540 LPRINT SPC(0);
2550 GOTO 2610
2560 LPRINT SPC(1);
2570 IF LJ < 3 THEN 2600
2580 PRINT SPC(0);
2590 GOTO 2610
2600 PRINT SPC(1);
2610 RETURN
2620 CLS:END
```

```
JANEIRO/ 1970
------------------------------------
DO   SE   TE   QA   QI   SE   SA
                    1    2    3
4    5    6    7    8    9    10
11   12   13   14   15   16   17
18   19   20   21   22   23   24
25   26   27   28   29   30   31

FEVEREIRO/ 1970
------------------------------------
DO   SE   TE   QA   QI   SE   SA

1    2    3    4    5    6    7
8    9    10   11   12   13   14
15   16   17   18   19   20   21
22   23   24   25   26   27   28

MARCO/ 1970
------------------------------------
DO   SE   TE   QA   QI   SE   SA

1    2    3    4    5    6    7
8    9    10   11   12   13   14
15   16   17   18   19   20   21
22   23   24   25   26   27   28
29   30   31

ABRIL/ 1970
------------------------------------
DO   SE   TE   QA   QI   SE   SA
               1    2    3    4
5    6    7    8    9    10   11
12   13   14   15   16   17   18
19   20   21   22   23   24   25
26   27   28   29   30

MAIO/ 1970
------------------------------------
DO   SE   TE   QA   QI   SE   SA
                         1    2
3    4    5    6    7    8    9
10   11   12   13   14   15   16
17   18   19   20   21   22   23
24   25   26   27   28   29   30
31

JUNHO/ 1970
------------------------------------
DO   SE   TE   QA   QI   SE   SA
     1    2    3    4    5    6
7    8    9    10   11   12   13
14   15   16   17   18   19   20
21   22   23   24   25   26   27
28   29   30

JULHO/ 1970
------------------------------------
DO   SE   TE   QA   QI   SE   SA
               1    2    3    4
5    6    7    8    9    10   11
12   13   14   15   16   17   18
19   20   21   22   23   24   25
26   27   28   29   30   31

AGOSTO/ 1970
------------------------------------
DO   SE   TE   QA   QI   SE   SA
                              1
2    3    4    5    6    7    8
9    10   11   12   13   14   15
16   17   18   19   20   21   22
23   24   25   26   27   28   29
30   31

SETEMBRO/ 1970
------------------------------------
DO   SE   TE   QA   QI   SE   SA
          1    2    3    4    5
6    7    8    9    10   11   12
13   14   15   16   17   18   19
20   21   22   23   24   25   26
27   28   29   30

OUTUBRO/ 1970
------------------------------------
DO   SE   TE   QA   QI   SE   SA
                    1    2    3
4    5    6    7    8    9    10
11   12   13   14   15   16   17
18   19   20   21   22   23   24
25   26   27   28   29   30   31

NOVEMBRO/ 1970
------------------------------------
DO   SE   TE   QA   QI   SE   SA

1    2    3    4    5    6    7
8    9    10   11   12   13   14
15   16   17   18   19   20   21
22   23   24   25   26   27   28
29   30

DEZEMBRO/ 1970
------------------------------------
DO   SE   TE   QA   QI   SE   SA
          1    2    3    4    5
6    7    8    9    10   11   12
13   14   15   16   17   18   19
20   21   22   23   24   25   26
27   28   29   30   31
```

Figura 5. Programa Calendário: exemplo de impressão do calendário de 1970.


```
JANEIRO/ 2000
------------------------------------
DO   SE   TE   QA   QI   SE   SA
                              1
2    3    4    5    6    7    8
9    10   11   12   13   14   15
16   17   18   19   20   21   22
23   24   25   26   27   28   29
30   31

FEVEREIRO/ 2000
------------------------------------
DO   SE   TE   QA   QI   SE   SA
          1    2    3    4    5
6    7    8    9    10   11   12
13   14   15   16   17   18   19
20   21   22   23   24   25   26
27   28   29

MARCO/ 2000
------------------------------------
DO   SE   TE   QA   QI   SE   SA
               1    2    3    4
5    6    7    8    9    10   11
12   13   14   15   16   17   18
19   20   21   22   23   24   25
26   27   28   29   30   31

ABRIL/ 2000
------------------------------------
DO   SE   TE   QA   QI   SE   SA
                              1
2    3    4    5    6    7    8
9    10   11   12   13   14   15
16   17   18   19   20   21   22
23   24   25   26   27   28   29
30

MAIO/ 2000
------------------------------------
DO   SE   TE   QA   QI   SE   SA
     1    2    3    4    5    6
7    8    9    10   11   12   13
14   15   16   17   18   19   20
21   22   23   24   25   26   27
28   29   30   31

JUNHO/ 2000
------------------------------------
DO   SE   TE   QA   QI   SE   SA
                    1    2    3
4    5    6    7    8    9    10
11   12   13   14   15   16   17
18   19   20   21   22   23   24
25   26   27   28   29   30

JULHO/ 2000
------------------------------------
DO   SE   TE   QA   QI   SE   SA
                              1
2    3    4    5    6    7    8
9    10   11   12   13   14   15
16   17   18   19   20   21   22
23   24   25   26   27   28   29
30   31

AGOSTO/ 2000
------------------------------------
DO   SE   TE   QA   QI   SE   SA
          1    2    3    4    5
6    7    8    9    10   11   12
13   14   15   16   17   18   19
20   21   22   23   24   25   26
27   28   29   30   31

SETEMBRO/ 2000
------------------------------------
DO   SE   TE   QA   QI   SE   SA
                         1    2
3    4    5    6    7    8    9
10   11   12   13   14   15   16
17   18   19   20   21   22   23
24   25   26   27   28   29   30

OUTUBRO/ 2000
------------------------------------
DO   SE   TE   QA   QI   SE   SA

1    2    3    4    5    6    7
8    9    10   11   12   13   14
15   16   17   18   19   20   21
22   23   24   25   26   27   28
29   30   31

NOVEMBRO/ 2000
------------------------------------
DO   SE   TE   QA   QI   SE   SA
               1    2    3    4
5    6    7    8    9    10   11
12   13   14   15   16   17   18
19   20   21   22   23   24   25
26   27   28   29   30

DEZEMBRO/ 2000
------------------------------------
DO   SE   TE   QA   QI   SE   SA
                         1    2
3    4    5    6    7    8    9
10   11   12   13   14   15   16
17   18   19   20   21   22   23
24   25   26   27   28   29   30
31
```

Figura 6. Programa Calendário: exemplo de impressão do calendário do ano 2000.

MULTIPLIC./DIVISÃO

O programa **Multiplic./Divisão (MULDIV)**, parecido com o programa ADISUB, tem a função de melhorar seu conhecimento de multiplicação e divisão. O *layout* da tela e a lógica do programa são muito parecidos com o descrito no programa ADISUB.

Novamente, você pode ter um quadro em torno da conta, se quiser. Pode ainda escolher multiplicação, divisão, ou ambos (linha 240).

O primeiro e último números são selecionados randomicamente nas linhas 440 e 450, como no programa de adição/subtração. Nas linhas 460 - 520 a operação (divisão ou multiplicação) é determinada.

O método de mostrar os dois números é o mesmo que o do programa ADISUB: um caracter de cada vez é movido para as duas linhas no centro da tela e você é solicitado a responder na parte inferior da tela.

Este programa é particularmente bom para exercitá-lo em tabelas de multiplicação, já que podemos selecionar o final das faixas para ambos os números (primeiro e segundo). A propósito, você pode começar com o primeiro valor de 1 a 9 e o segundo de 1 a 12.

Tentando responder o mais rápido possível, adultos também podem melhorar seus conhecimentos com MULDIV. Lembre-se de que ambos os números podem ter a faixa de operação até 999.

```
10 REM ************************
20 REM *                      *
30 REM * MULTIPLIC./DIVISAO   *
40 REM *                      *
50 REM * DIREITOS AUTORAIS    *
60 REM *                      *
70 REM *     NELSON ALVES     *
80 REM *                      *
90 REM *                      *
100 REM ***********************
110 KEY OFF
120 NEG$ = "N"
130 CLS
140 INPUT "NUMERO RANDOMICO (1 A 9999)";X
150 INPUT "COM REQUADRO (S/N)";SN$
160 INPUT "PRIMEIRO NUMERO DE 1 A (2/999)";T1
170 IF T1 < 2 THEN 160
180 IF T1 > 999 THEN 160
190 INPUT "SEGUNDO NUMERO DE 1 A (2/999)";B1
```



```
200 IF B1 < 2 THEN 190
210 IF B1 > 999 THEN 190
220 ERS = 0
230 C1 = 0
240 INPUT "DIVISAO, MULTIPLIC., AMBOS (D,M,A)";ASB$
250 IF ASB$ = "D" THEN 290
260 IF ASB$ = "M" THEN 290
270 IF ASB$ = "A" THEN 290
280 GOTO 240
290 CLS: IF SN$ = "N" THEN 440
300 R = 13
310 FOR C = 5 TO 15
320 LOCATE R,C
330 PRINT CHR$(189);
340 LOCATE (R+10),C
350 PRINT CHR$(189);
360 NEXT C
370 C = 5
380 FOR R = 13 TO 23
390 LOCATE R,C
400 PRINT CHR$(189);
410 LOCATE R,(C+10)
420 PRINT CHR$(189);
430 NEXT R
440 N1 = INT(T1*RND(X))+1
450 N2 = INT(B1*RND(X))+1
460 IF NEG$ <> "N" THEN 510
470 IF N1 > N2 THEN 510
510 IF ASB$ = "D" THEN 570
520 IF ASB$ = "M" THEN 550
530 O1 = INT(2*RND(X))+1
540 GOTO 580
550 O1 = 1
560 GOTO 580
570 O1 = 2
580 IF O1 = 1 THEN 590
590 N1 = N1*N2
595 N1$ = STR$(N1)
600 N2$ = STR$(N2)
610 IF O1 = 1 THEN 640
620 OP$ = "/"
630 GOTO 650
640 OP$ = "X"
650 L1 = LEN(N1$)
660 L2 = LEN(N2$)
```

```
670 J = 20
680 FOR I = L1 TO 1 STEP -1
690 J = J-1
700 LOCATE J,8
710 PRINT MID$(N1$,I,1);
720 NEXT I
730 J = 20
740 FOR I = L2 TO 1 STEP -1
750 J = J-1
760 LOCATE J,9
770 PRINT MID$(N2$,I,1)
780 NEXT I
790 LOCATE 16,9
800 PRINT OP$;
810 FOR I = 16 TO 19
820 LOCATE I,10
830 PRINT "_";
840 NEXT I
850 LOCATE 0,23
860 INPUT "DIGITE (99999 PARA FINALIZAR): ";X1
870 IF X1 = 99999! THEN 1030
880 IF O1 = 2 THEN 910
890 IF X1 = N1*N2 THEN 990
900 GOTO 920
910 IF X1 = N1/N2 THEN 990
920 CLS
930 IF O1 = 2 THEN 960
940 PRINT "ERRADO ";N1; " x ";N2; " = ";N1*N2
950 GOTO 970
960 PRINT "ERRADO ";N1; " / ";N2; " = "; N1/N2
970 ERS = ERS+1
980 GOTO 1010
990 C1 = C1+1
1000 GOTO 290
1010 INPUT X$
1020 GOTO 290
1030 CLS
1040 PRINT "**************************************"
1050 PRINT "*"; TAB(38);"*"
1060 PRINT "*             CORRETO = ";C1;TAB(38); "*"
1070 PRINT "*"; TAB(38); "*"
1080 PRINT "*             ERROS = "; ERS; TAB(38); "*"
1090 PRINT "*"; TAB(38); "*"
1100 IF C1 > 0 THEN 1130
1110 IF ERS > 0 THEN 1130
```



```
1120 ERS = 1
1130 PRINT "*              ESCORE = "; INT(C1/(C1+
   ERS)*100); "%"; TAB(38); "*"
1140 PRINT "*"; TAB(38); "*"
1150 PRINT "**************************************"
1160 PRINT
1170 PRINT
1180 PRINT
1190 PRINT
1200 END
```

GRÁFICOS DE BARRAS

O programa **Gráfico de Barras (GRÁFICO)** desenha um gráfico de barras horizontal para qualquer conjunto de figuras. Até 10 barras podem ser mostradas em um gráfico. Cada barra pode ser desenhada e ter um nome que será impresso à esquerda da barra. O nome pode conter até 9 caracteres (o título do elemento na linha 280).

No início do programa, você é solicitado a dar o nome do gráfico. Este título pode ter até 18 caracteres (linhas 190 - 200). Você pode também precisar identificar o valor máximo que quer que o gráfico seja capaz de desenhar (linha 170). Note que digitar o valor máximo como 0 termina o programa (linha 180).

Após ter ajustado o valor máximo, comece a inserir os valores dos elementos 1 até o número de elementos desejado. Digite os valores até digitar FIM para o nome do elemento; neste momento a seção de entrada de dados termina (linhas 260 - 350). O valor máximo previamente digitado é então dividido por 12 para ter o valor de um quadrado ou bloco do gráfico. (Veja a linha 360). Em outras palavras, se qualquer elemento tiver um valor máximo (eles todos podem ter valores menores ou o mesmo, mas não maior), o comprimento de cada barra pode ser de 12 blocos.

As linhas 400 - 520 então mostram o gráfico na tela e/ou na impressora (se você responder S na linha 220). O título é impresso no topo, e o nome de cada elemento do gráfico precede a barra para o seu valor. Se desenhado também na impressora, *Os* são usados no lugar dos blocos na tela. Quando o gráfico está completo, as opções são **FIM** (**F**), **IMPRIMIR** (**I**) ou continuar com um novo gráfico **(RETURN)**.

```
10 REM  ************************
20 REM  *                      *
30 REM  * GRAFICO  DE  BARRAS  *
40 REM  *                      *
50 REM  * DIREITOS AUTORAIS    *
60 REM  *                      *
70 REM  *    NELSON ALVES      *
80 REM  *                      *
90 REM  *                      *
100 REM ************************
110 KEY OFF
130 DIM V(10)
140 DIM N$(10)
150 CLS
160 CTR = 0
170 INPUT "VALOR MAXIMO: ";MA1
```



```
180 IF MA1 = 0 THEN 700
190 INPUT "NOME DO GRAFICO (18 CARACS.): ";GN$
200 IF LEN(GN$) > 18 THEN 190
210 ISN$ = "N"
220 INPUT "IMPRESSORA (S/N) ";ISN$
230 IF ISN$ = "S" THEN 250
240 IF ISN$ <> "N" THEN 220
250 CLS
260 FOR I = 1 TO 10
270 PRINT "NO. DO ELEMENTO: ";I
280 INPUT "TITULO DO ELEMENTO (9 CARACS.): ";N$(I)
290 IF LEN(N$(I)) > 9 THEN 280
300 IF N$(I) = "FIM" OR N$(I) = "" THEN 360
310 CTR = CTR + 1
320 INPUT "VALOR DO ELEMENTO ";V(I)
330 IF V(I) > MA1 THEN 320
340 IF V(I) < 0 THEN 320
350 NEXT I
360 GRAFO = INT(MA1/12)
370 CLS
380 FOR I = 1 TO CTR
390 IF ISN$ <> "S" THEN 420
400 LPRINT N$(I);SPC(9-LEN(N$(I)));
410 GOTO 430
420 PRINT N$(I);SPC(9-LEN(N$(I)));
430 V2 = INT(V(I)/GRAFO)
440 IF V2 = 0 THEN 550
450 FOR J = 1 TO V2
460 IF ISN$ <> "S" THEN 490
470 LPRINT CHR$(219);
480 GOTO 500
490 PRINT CHR$(219);
500 NEXT J
510 IF ISN$ <> "S" THEN 550
520 LPRINT
530 LPRINT SPC(9);V(I)
540 GOTO 570
550 PRINT
560 PRINT SPC(9);V(I)
570 NEXT I
580 IF ISN$ <> "S" THEN 630
590 LPRINT
600 LPRINT CHR$(219); " = ";GRAFO;SPC(5);GN$
610 GOTO 650
620 PRINT
```

```
630 PRINT CHR$(219); " = ";GRAFO;SPC(5);GN$
640 PRINT: GOTO 660
650 LPRINT
660 INPUT "F=FIM/I=IMPRIMIR/RETURN=CONTINUAR ";E$
670 IF E$ = "F" THEN 700
680 IF E$ = "I" THEN ISN$ = "S": GOTO 370
690 GOTO 150
700 CLS
710 END
```


ANÁLISE DE LÓGICA

O programa **Análise de Lógica (ANALOG)** é um programa onde você tenta "chutar" um número de três dígitos baseado em dicas que obtém de "chutes" anteriores.

Primeiro, o programa seleciona um número randômico — de 1 a 999 (linha 230). Nas linhas 240 - 320, um número de um ou dois dígitos teve zeros a esquerda adicionados à série de caracteres representada por X$ (linha 240). Assim, o número 9 é 009, e 98 é 098. A variável X1$, X2$ e X3$ contém o primeiro, segundo e terceiro dígito que será comparado com o número chutado.O número de tentativas que você fez é incrementado na linha 360 antes do chute de três dígitos da linha 380.

Se você falhar na digitação dos três dígitos, a linha 400 retornará para a linha 380 para tentar novamente. (Para chutar o número 3, você deve digitar 003). Y1$, Y2$ e Y3$ recebem os valores do número de três dígitos para ser comparado com os dígitos atuais. As linhas 490 - 560 verificam se os números corretos estão nas posições corretas. Se todos os três números estiverem corretos, você chutou um número randômico e o controle será transferido para a linha 550 até a linha 660.

Caso contrário, as linhas 570 - 620 verificam cada dígito que você chutou contra cada dígito do número e diz quantos dígitos do seu chute estão corretos (mas não necessariamente na posição correta). Usando a informação acumulada, você deve tentar chutar o número em menos tentativas quanto for possível.

```
10 REM ************************
20 REM *                      *
30 REM * ANALISE  DE  LOGICA *
40 REM *                      *
50 REM *  DIREITOS AUTORAIS  *
60 REM *                      *
70 REM *    NELSON ALVES      *
80 REM *                      *
90 REM *                      *
100 REM ************************
110 KEY OFF
120 CLS
130 INPUT "DIGITE QUALQUER NUMERO ";A1
140 CLS
150 PRINT TAB(7);"************************"
160 PRINT TAB(7);"*                      *"
170 PRINT TAB(7);"*    ANALISE LOGICA    *"
180 PRINT TAB(7);"*                      *"
190 PRINT TAB(7);"************************"
```

```
200 LOCATE 7,15:INPUT "JOGAR(J) OU FIM(F) ";JF$
210 IF JF$ = "F" THEN 840
220 CLS
230 X = INT(RND(A1)*999)+1
240 X$ = STR$(X)
250 CTR = 0
260 LX = LEN(X$)
270 ON LX GOTO 300,300,320,330
280 PRINT LX
290 GOTO 200
300 X$ = "OO" + X$
310 GOTO 330
320 X$ = "O"+X$
330 X1$ = MID$(X$,2,1)
340 X2$ = MID$(X$,3,1)
350 X3$ = MID$(X$,4,1)
360 CTR = CTR+1
370 PRINT "TENTATIVA NO.: ";CTR
380 INPUT "TENTE COM 3 DIGITOS ";Y$
390 LY = LEN(Y$)
400 IF LY <> 3 THEN 380
410 Y1$ = MID$(Y$,1,1)
420 Y2$ = MID$(Y$,2,1)
430 Y3$ = MID$(Y$,3,1)
440 CNUM = 0
450 CP = 0
460 X1F = 0
470 X2F = 0
480 X3F = 0
490 IF X1$ <> Y1$ THEN 510
500 CP = CP+1
510 IF X2$ <> Y2$ THEN 530
520 CP = CP+1
530 IF X3$ <> Y3$ THEN 560
540 CP = CP+1
550 IF CP = 3 THEN 650
560 Z$ = Y1$
570 GOSUB 690
580 Z$ = Y2$
590 GOSUB 690
600 Z$ = Y3$
610 GOSUB 690
620 PRINT "VOCE TEM...";CP;" (POSICAO CORRETA"
630 PRINT "E...";CNUM;" (DIGITO(S) CORRETO(S)"
640 GOTO 360
```



```
650 PRINT:PRINT:PRINT:PRINT "CERTO!!!";CHR$(7);", ELE E ";X$
660 PRINT:PRINT "*************************************"
670 LOCATE 0,23:INPUT "DIGITE (J)OGAR / (F)IM ";JF$
680 IF JF$ = "F" THEN GOTO 840 ELSE 220
690 IF X1F = 1 THEN 740
700 IF Z$ <> X1$ THEN 740
710 X1F = 1
720 CNUM = CNUM+1
730 GOTO 830
740 IF X2F = 1 THEN 790
750 IF Z$ <> X2$ THEN 790
760 X2F = 1
770 CNUM = CNUM+1
780 GOTO 830
790 IF X3F = 1 THEN 830
800 IF Z$ <> X3$ THEN 830
810 X3F = 1
820 CNUM = CNUM+1
830 RETURN
840 CLS: END
```

ANÁLISE DE LÓGICA

O programa **Análise de Lógica (ANALO1)** embora seja parecido com o ANALOG requer que você "chute" quatro dígitos númericos ao invés de um número com três dígitos.

Na linha 230, um número randômico entre 1 e 999 é selecionado. As linhas 260 - 330 adicionam zeros à esquerda de números menores do que 1.000.

O número que você chutou na linha 400 é comparado com o número a ser tentado, assim como no programa ANALOG. Obviamente você precisará de mais tentativas para descobrir um número de quatro dígitos que um número de três dígitos.

```
10 REM  ************************
20 REM  *                      *
30 REM  * ANALISE  DE  LOGICA *
40 REM  *                      *
50 REM  *  DIREITOS AUTORAIS   *
60 REM  *                      *
70 REM  *     NELSON ALVES     *
80 REM  *                      *
90 REM  *                      *
100 REM ************************
110 KEY OFF
120 CLS
130 INPUT "DIGITE QUALQUER NUMERO ";A1
140 CLS
150 PRINT TAB(7);"*************************"
160 PRINT TAB(7);"*                       *"
170 PRINT TAB(7);"*   ANALISE LOGICA      *"
180 PRINT TAB(7);"*                       *"
190 PRINT TAB(7);"*************************"
200 LOCATE 7,15:INPUT "JOGAR(J) OU FIM(F) ";JF$
210 IF JF$ = "F" THEN 970
220 CLS
230 X = INT(RND(A1)*9999)+1
240 X$ = STR$(X)
250 CTR = 0
260 LX = LEN(X$)
270 ON LX GOTO 280,280,310,330,340
280 GOTO 200
290 X$ = "000" + X$
300 GOTO 340
310 X$ = "00" + X$
```



```
320 GOTO 340
330 X$ = "0"+X$
340 X1$ = MID$(X$,2,1)
350 X2$ = MID$(X$,3,1)
360 X3$ = MID$(X$,4,1)
370 X4$ = MID$(X$,5,1)
380 CTR = CTR+1
390 PRINT "TENTATIVA NO.: ";CTR
400 INPUT "TENTE COM 4 DIGITOS ";Y$
410 LY = LEN(Y$)
420 IF LY <> 4 THEN 400
430 Y1$ = MID$(Y$,1,1)
440 Y2$ = MID$(Y$,2,1)
450 Y3$ = MID$(Y$,3,1)
460 Y4$ = MID$(Y$,4,1)
470 CNUM = 0
480 CP = 0
490 X1F = 0
500 X2F = 0
510 X3F = 0
520 X4F = 0
530 IF X1$ <> Y1$ THEN 550
540 CP = CP+1
550 IF X2$ <> Y2$ THEN 570
560 CP = CP+1
570 IF X3$ <> Y3$ THEN 590
580 CP = CP+1
590 IF X4$ <> Y4$ THEN 620
600 CP = CP + 1
610 IF CP = 4 THEN 730
620 Z$ = Y1$
630 GOSUB 770
640 Z$ = Y2$
650 GOSUB 770
660 Z$ = Y3$
670 GOSUB 770
680 Z$ = Y4$
690 GOSUB 770
700 PRINT "VOCE TEM...";CP;" (POSICAO CORRETA"
710 PRINT "E...";CNUM;" (DIGITO(S) CORRETO(S)"
720 GOTO 380
730 PRINT:PRINT:PRINT:PRINT "CERTO!!!";CHR$(7);", ELE E ";X$
740 PRINT:PRINT "****************************************"
750 LOCATE 0,23:INPUT "DIGITE (J)OGAR / (F)IM ";JF$
760 IF JF$ = "F" THEN GOTO 970 ELSE 220
```

```
770 IF X1F = 1 THEN 820
780 IF Z$ <> X1$ THEN 820
790 X1F = 1
800 CNUM = CNUM+1
810 GOTO 960
820 IF X2F = 1 THEN 870
830 IF Z$ <> X2$ THEN 870
840 X2F = 1
850 CNUM = CNUM+1
860 GOTO 960
870 IF X3F = 1 THEN 920
880 IF Z$ <> X3$ THEN 920
890 X3F = 1
900 CNUM = CNUM+1
910 GOTO 960
920 IF X4F = 1 THEN 960
930 IF Z$ <> X4$ THEN 960
940 X4F = 1
950 CNUM = CNUM + 1
960 RETURN
970 CLS: END
```


AMORTIZAÇÃO

O programa **Amortização (AMORT)** gera o montante do pagamento mensal necessário para amortizar um empréstimo sobre qualquer número de meses a qualquer taxa de juros desejada. Na linha 140 o montante principal do empréstimo é colocado e as linhas 160 - 180 pedem a taxa de juros anual e o número de meses necessários para o empréstimo ser pago. O juro mensal é computado como a taxa anual dividida por 12 na linha 190. Como podemos ver, a linha 170 divide a taxa de juros digitada (digamos 12, por exemplo) por 100, dando .12 para cálculos. O montante do pagamento é então derivado da fórmula na linha 200 e as linhas 210 - 270 arredondam o pagamento para o inteiro mais próximo do centavo.

Nesta hora você é perguntado se quer ver o cronograma detalhado do pagamento com o balanço do declínio do empréstimo. Se você só quer comparar as diferenças nos pagamentos entre várias taxas de juros e/ou periodos de pagamento, pode não querer ver o relatório detalhado de pagamentos. Entretanto, se quiser ter uma idéia do montante de juros pagos a cada ano (para propósitos de taxa), poderá ver os pagamentos e balanços exatos.

Além disso, você pode ver o cronograma de pagamentos e balanços na tela; a tela pára quando cada uma é preenchida (pressione **RETURN** para continuar na próxima tela). Ou, se tiver uma impressora, você poderá gerar uma listagem da mesma informação.

Quando um cronograma for completado, você poderá então continuar com outro cronograma ou finalizar o programa digitando 0 para o montante principal na linha 140.

```
10 REM ***********************
20 REM *                     *
30 REM *    AMORTIZACAO      *
40 REM *                     *
50 REM *  DIREITOS AUTORAIS  *
60 REM *                     *
70 REM *    NELSON ALVES     *
80 REM *                     *
90 REM *                     *
100 REM ***********************
110 KEY OFF
130 CLS
140 INPUT "PRINCIPAL ";PRIN
150 IF PRIN <= 0 THEN 780
160 INPUT "TAXA DE JUROS ANUAL (%) ";TJ
```

```
170 TJ = TJ / 100
180 INPUT "NO. DE MESES ";N
190 IM = TJ / 12
200 P = PRIN*IM*(1+IM)^N/(((1+IM)^N)-1)
210 P2 = P*100
220 P3 = INT(P2)+.5
230 IF P2 >= P3 THEN 260
240 P2 = INT(P2)
250 GOTO 270
260 P2 = INT(P2) + 1
270 PAMT = P2/100
280 PRINT "O PAGAMENTO MENSAL E DE ";PAMT
290 INPUT "PLANILHA DETALHADA (S/N) ";SN$
300 IF SN$ = "N" THEN 170
310 IF SN$ = "S" THEN 330
320 GOTO 290
330 INPUT "IMPRESSORA (S/N) ";ISN$
340 INPUT "CALCULAR O TOTAL (S/N) ";SN$
350 IF SN$ = "S" THEN 390
360 IF SN$ = "N" THEN 380
370 GOTO 340
380 INPUT "PGTO MENSAL DESEJADO ";PAMT
390 IF ISN$ <> "S" THEN 480
400 LPRINT CHR$(12)
410 LPRINT "PRINCIPAL: ";PRIN
420 LPRINT "JUROS ANUAIS: ";TJ
430 LPRINT "TOTAL PAGO: ";PAMT
440 LPRINT
450 LPRINT "PRINCIPAL";SPC(8);"JUROS";SPC(11);
460 LPRINT "P/ PRINCIPAL";SPC(4);"TOTAL PAGO";SPC(5);"MES"
470 LPRINT
480 CTRP = 0
490 FOR I = 1 TO N
500 I1 = PRIN * IM
510 I1 = (INT((I1+5E-03)*100)/100)
520 IF I < N THEN 540
530 PAMT = PRIN + I1
540 TPRIN = PAMT - I1
550 TPRIN = (INT((TPRIN+5E-03)*100)/100)
560 PRIN = PRIN - TPRIN
570 PRIN = (INT((PRIN+5E-03)*100)/100)
580 IF PRIN >= 0 THEN 630
590 PRIN = PRIN + TPRIN
600 PAMT = PRIN
610 TPRIN = PAMT - I1
```



```
620 PRIN = 0
630 PRINT "PRIN = ";PRIN;TAB(18);"JUROS = ";I1
640 PRINT "P/ PRIN = ";TPRIN; TAB(18);"PGTO = ";PAMT
650 PRINT "******************************"
660 CTRP = CTRP + 1
670 IF ISN$ = "S" THEN 720
680 IF CTRP < 7 THEN 710
690 INPUT "DIGITE RETURN PARA CONTINUAR ";X$
700 CTRP = 0
710 IF ISN$ <> "S" THEN 750
720 LPRINT PRIN;SPC(15 - LEN(STR$(PRIN)));I1;SPC(15-LEN
    (STR$(I1)));TPRIN;
730 LPRINT SPC(15 - LEN(STR$(TPRIN)));PAMT;
    SPC(15-LEN(STR$(PAMT)));I
740 NEXT I
750 INPUT "CONTINUAR (S/N) ";SN$
760 IF SN$ <> "S" THEN 780
770 GOTO 130
780 CLS: END
```

```
PRINCIPAL:   10000
JUROS ANUAIS:   .23
TOTAL PAGO:   523.73
```

PRINCIPAL	JUROS	P/ PRINCIPAL	TOTAL PAGO	MES
9667.94	191.67	332.06	523.73	1
9329.51	185.3	338.43	523.73	2
8984.6	178.82	344.91	523.73	3
8633.07	172.2	351.53	523.73	4
8274.81	165.47	358.26	523.73	5
7909.68	158.6	365.13	523.73	6
7537.55	151.6	372.13	523.73	7
7158.29	144.47	379.26	523.73	8
6771.76	137.2	386.53	523.73	9
6377.82	129.79	393.94	523.73	10
5976.33	122.24	401.49	523.73	11
5567.15	114.55	409.18	523.73	12
5150.12	106.7	417.03	523.73	13
4725.1	98.71	425.02	523.73	14
4291.93	90.56	433.17	523.73	15
3850.46	82.26	441.47	523.73	16
3400.53	73.8	449.93	523.73	17
2941.98	65.18	458.55	523.73	18
2474.64	56.39	467.34	523.73	19
1998.34	47.43	476.3	523.73	20
1512.91	38.3	485.43	523.73	21
1018.18	29	494.73	523.73	22
513.97	19.52	504.21	523.73	23
0	9.85	513.97	523.82	24

Figura 7. Programa Amortização: exemplo de saída do programa.

INVESTIMENTOS

O programa **Investimentos (INVEST)** permite que você analize várias poupanças e investimentos. Basicamente, dois tipos maiores de investimentos são analizados: um investimento parcial e um pagamento anual que adiciona a uma conta de poupança uma base regular. Existem duas espécies de anuidades: uma anuidade ordinária (simples) e uma anuidade devida. Em uma anuidade ordinária, os depósitos são feitos no final do período, o que significa que nenhum juro é recebido no último montante pago. Em uma anuidade devida, depósitos são feitos no início de cada período e o juro é recebido no último pagamento efetuado. A diferença entre as duas anuidades é essencialmente a diferença de tempo.

O menu da tela nas linhas 150 - 340 descreve os comandos que podem ser utilizados:

1. IM (um investimento parcial composto mensalmente) analiza quanto um investimento de, digamos Cz$ 500.000,00, pode crescer em um determinado número de meses.
2. IA (im investimento parcial composto anualmente) faz a mesma coisa sobre um período de anos.
3. AMO (uma anuidade ordinária) diz a você quanto teria se tivesse investido, digamos Cz$ 5.00,00, todo mês por um período de 36 meses.
4. AO (anuidade ordinária) analiza o pagamento de uma anuidade feito, anualmente, por um certo número de anos.
5. MD (mensalidade devida) faz o mesmo que AMO, mas o pagamento é feito no início de cada período de juros, assim mais juros é ganho.
6. AD (anuidade devida) é similar a AO, mas o pagamento é feito no início de cada período de juros.

Cada um dos seis diferentes comandos solicitará uma quantia de cruzados para investimento ou pagamento (dependendo se for para um investimento ou anuidade). Uma taxa de juros deve ser digitada. Por exemplo, uma taxa anual de juros de 12.75% deve ser digitada como 12.75, e 13% como 13. O valor após o período de tempo requerido é então exibido. Se você quer um relatório detalhado, pode tê-lo mês-a-mês ou ano-a-ano.

Comandos específicos estão incluídos nas linhas 1.000 - 1.010, que são uma simples computação do investimento, baseado na fórmula financeira: valor futuro = principal vezes (1+a taxa de juro) elevado à potência do número de pagamentos. Na linha 1.050, o valor futuro é então arredondado para os centavos mais próximos. As fórmulas das anuidades estão nas linhas 1.760 - 1.840 e nas linhas 2.060 - 2.090.

Este programa, assim como outros programas neste livro, podem utilizar impressora. Se você tiver uma, pode imprimir cronogramas de análise de poupanças e anuidades.

```
10 REM *************************
20 REM *                       *
30 REM *    INVESTIMENTOS      *
40 REM *                       *
50 REM *  DIREITOS AUTORAIS    *
60 REM *                       *
70 REM *     NELSON ALVES      *
80 REM *                       *
90 REM *                       *
100 REM ************************
110 KEY OFF
130 IPR$ = "N"
140 CLS
150 PRINT
160 PRINT SPC(5);"******************************"
170 PRINT SPC(5);"*                            *"
180 PRINT SPC(5);"*  ANALISE DE INVESTIMENTOS  *"
190 PRINT SPC(5);"*                            *"
200 PRINT SPC(5);"******************************"
210 PRINT
220 PRINT
230 PRINT "    IM  INVESTIMENTO COMPOSTO MENSAL"
240 PRINT "    IA  INVESTIMENTO COMPOSTO ANUAL"
250 PRINT "   AMO INVESTIMENTO MENSAL (ORDINARIO)"
260 PRINT "    AO  INVESTIMENTO ANUAL (ORDINARIO)"
270 PRINT "    MD  MENSALIDADE DEVIDA"
280 PRINT "    AD  ANUIDADE DEVIDA"
290 PRINT "    AI  ATIVAR A IMPRESSORA"
300 PRINT "    DI  DESATIVAR A IMPRESSORA"
310 PRINT "   FIM FINALIZA O PROGRAMA"
320 PRINT
330 PRINT
340 PRINT
350 REM ANUIDADES ORDINARIAS NAO ADMITEM
360 REM JUROS NO ULTIMO PAGAMENTO
370 REM PORQUE OS PAGAMENTOS SAO APROPRIADOS
380 REM NO FINAL DO PERIODO
390 REM ************************************
400 INPUT "FACA A SUA OPCAO ";S$
410 IF S$ = "FIM" THEN 2520
```



```
420 IF S$ = "IM" THEN 600
430 IF S$ = "IA" THEN 960
440 IF S$ = "AMO" THEN 1340
450 IF S$ = "AO" THEN 1940
460 IF S$ = "MD" THEN 1340
470 IF S$ = "AD" THEN 1940
480 IF S$ = "AI" THEN 520
490 IF S$ = "DI" THEN 560
500 GOTO 140
510 REM ***********************************
520 IF IPR$ = "S" THEN 540
530 IPR$ = "S"
540 GOTO 140
550 REM ***********************************
560 IF IPR$ = "N" THEN 580
570 IPR$ = "N"
580 GOTO 140
590 REM ***********************************
600 CLS
610 PRINT "INVESTIMENTO COMPOSTO MENSAL"
620 INPUT "TOTAL DO INVESTIMENTO ";TI
630 INPUT "TAXA DE JURO ANUAL ";JA
640 IF JA >= 1 THEN 670
650 PRINT "DIGITE 12% COMO 12"
660 GOTO 630
670 MI = JA / 1200
680 INPUT "NO. DE MESES ";ME
690 FV = TI*(1+MI)^MO
700 FV = (INT((FV+5E-03)*100)/100)
710 PRINT "VALOR APOS ";ME;"MES(ES): ";FV
720 PRINT
730 IF IPR$ <> "S" THEN 790
740 LPRINT "INVESTIMENTO COMPOSTO MENSAL"
750 LPRINT "TOTAL DO INVESTIMENTO: ";TI
760 LPRINT "TAXA DE JURO ANUAL: ";JA
770 LPRINT "NO. DE MESES: ";ME
780 LPRINT
790 INPUT "IMPRIMIR DETALHES MENSAIS (S/N) ";SN$
800 IF SN$ <> "S" THEN 920
810 FOR I = 1 TO ME
820 FV = TI*(1+MI)
830 FV = (INT((FV+5E-03)*100)/100)
840 IF IPR$ <> "S" THEN 880
850 LPRINT "MES ";I;
860 LPRINT SPC(10 - LEN(STR$(I)));FV
```

```
870 GOTO 900
880 PRINT "MES ";I;
890 PRINT SPC(10 - LEN(STR$(I)));FV
900 TI = FV
910 NEXT I
920 PRINT
930 INPUT "TECLE RETURN PARA CONTINUAR ";X$
940 GOTO 140
950 REM ************************************
960 CLS
970 INPUT "TOTAL DO INVESTIMENTO ";TI
980 INPUT "TAXA DE JURO ANUAL ";JA
990 IF JA > 1 THEN 1020
1000 PRINT "DIGITE 12% COMO 12"
1010 GOTO 980
1020 JA = JA /100
1030 INPUT "NO. DE ANOS ";AN
1040 FV = TI*(1+JA)^AN
1050 FV = (INT((FV+5E-03)*100)/100)
1060 PRINT "VALOR APOS ";AN;" ANO(S)";FV
1070 PRINT "TOTAL DO INVESTIMENTO: ";TI
1080 PRINT "TAXA ANUAL DE JURO: ";JA
1090 PRINT
1100 IF IPR$ <> "S" THEN 1170
1110 LPRINT "INVESTIMENTO COMPOSTO ANUAL"
1120 LPRINT "TOTAL DO INVESTIMENTO: ";TI
1130 LPRINT "TAXA DE JURO ANUAL: ";JA
1140 LPRINT "NO. DE ANOS: ";AN
1150 LPRINT "VALOR APOS ";AN;" ANOS";FV
1160 LPRINT
1170 INPUT "IMPRIMIR DETALHES ANUAIS (S/N) ";SN$
1180 IF SN$ <> "S" THEN 1300
1190 FOR I = 1 TO AN
1200 FV = TI*(1+JA)
1210 FV = (INT((FV+5E-03)*100)/100)
1220 IF IPR$ <> "S" THEN 1260
1230 LPRINT "ANO ";I;
1240 LPRINT SPC(10 - LEN(STR$(I)));FV
1250 GOTO 1280
1260 PRINT "ANO ";I;
1270 PRINT SPC(10 - LEN(STR$(I)));FV
1280 TI = FV
1290 NEXT I
1300 PRINT
1310 INPUT "TECLE RETURN PARA CONTINUAR ";X$
```



```
1320 GOTO 140
1330 REM ************************************
1340 CLS
1350 IF S$ = "AMO" THEN 1380
1360 PRINT "MENSALIDADE DEVIDA"
1370 GOTO 1390
1380 PRINT "INVESTIMENTO MENSAL (ORDINARIO)"
1390 INPUT "TOTAL DO INVESTIMENTO MENSAL ";TIM
1400 INPUT "TAXA DE JURO ANUAL ";JA
1410 IF JA >= 1 THEN 1440
1420 PRINT "DIGITE 12% COMO 12"
1430 GOTO 1400
1440 MI = JA / 1200
1450 INPUT "NO. DE MESES PAGOS ";ME
1460 AMT = (TIM/MI)*((1+MI)^ME-1)
1470 IF S$ = "AMO" THEN 1490
1480 AMT = AMT*(1+MI)
1490 AMT = (INT((AMT+5E-03)*100)/100)
1500 CLS
1510 PRINT
1520 IF S$ = "AMO" THEN 1650
1530 IF IPR$ <> "S" THEN 1560
1540 LPRINT "MENSALIDADE DEVIDA"
1550 GOTO 1600
1560 PRINT "MENSALIDADE DEVIDA"
1570 GOTO 1670
1580 IF IMPR$ <> "S" THEN 1650
1590 LPRINT "INVESTIMENTO MENSAL (ORDINARIO)"
1600 LPRINT "TOTAL DO INVESTIMENTO MENSAL: ";TIM
1610 LPRINT "TAXA DE JURO ANUAL: ";JA
1620 LPRINT "TOTAL ANUAL: ";AMT
1630 LPRINT
1640 GOTO 1710
1650 IF IPR$ = "S" THEN 1590
1660 PRINT "INVESTIMENTO MENSAL (ORDINARIO)"
1670 PRINT "TOTAL DO INVESTIMENTO MENSAL: ";TIM
1680 PRINT "TAXA DE JURO ANUAL: ";JA
1690 PRINT "TOTAL ANUAL: ";AMT
1700 PRINT
1710 INPUT "IMPRIMIR DETALHES MENSAIS (S/N) ";SN$
1720 IF SN$ <> "S" THEN 1900
1730 AMT = 0
1740 FOR I = 1 TO ME
1750 IF S$ = "AMO" THEN 1770
1760 AMT = AMT + TIM
```

```
1770 FV = AMT*(1+MI)
1780 FX = FV
1790 FV = (INT((FV+5E-03)*100)/100)
1800 IF S$ = "MD" THEN 1840
1810 FV = FV+TIM
1820 AMT = FX+TIM
1830 GOTO 1850
1840 AMT = FX
1850 IF IPR$ <> "S" THEN 1880
1860 LPRINT "MES ";I;SPC(10-LEN(STR$(I)));FV
1870 GOTO 1890
1880 PRINT "MES ";I;SPC(10-LEN(STR$(I)));FV
1890 NEXT I
1900 PRINT
1910 INPUT "TECLE RETURN PARA CONTINUAR ";X$
1920 GOTO 140
1930 REM ***********************************
1940 CLS
1950 IF S$ = "AO" THEN 1980
1960 PRINT "ANUIDADE DEVIDA"
1970 GOTO 1990
1980 PRINT "INVESTIMENTO ANUAL (ORDINARIO)"
1990 INPUT "TOTAL DO INVESTIMENTO ANUAL ";TIM
2000 INPUT "TAXA DE JURO ANUAL ";JA
2010 IF JA > 1 THEN 2040
2020 PRINT "DIGITE 12% COMO 12"
2030 GOTO 2000
2040 JA = JA / 100
2050 INPUT "NO. DE ANOS PAGOS ";AN
2060 AMT = (TIM/JA)*((1+JA)^AN-1)
2070 IF S$ = "AO" THEN 2090
2080 AMT = AMT*(1+JA)
2090 AMT = (INT((AMT+5E-03)*100)/100)
2100 CLS
2110 PRINT
2120 IF S$ = "AO" THEN 2240
2130 IF IPR$ <> "S" THEN 2220
2140 LPRINT "ANUIDADE DEVIDA"
2150 GOTO 2170
2160 LPRINT "INVESTIMENTO ANUAL (ORDINARIO)"
2170 LPRINT "TOTAL DO INVESTIMENTO ANUAL: ";TIM
2180 LPRINT "TAXA DE JURO ANUAL: ";JA
2190 LPRINT "TOTAL DA ANUIDADE: ";AMT
2200 LPRINT
2210 GOTO 2300
```



```
2220 PRINT "ANUIDADE DEVIDA"
2230 GOTO 2260
2240 IF IPR$ = "S" THEN 2160
2250 PRINT "INVESTIMENTO ANUAL (ORDINARIO)"
2260 PRINT "TOTAL DO IN+ESTIMENTO ANUAL: ";TIM
2270 PRINT "TAXA DE JURO ANUAL: ";JA
2280 PRINT "TOTAL DA ANUIDADE: ";AMT
2290 PRINT
2300 INPUT "IMPRIMIR DETALHES ANUAIS (S/N) ";SN$
2310 IF SN$ <> "S" THEN 2490
2320 AMT = 0
2330 FOR I = 1 TO AN
2340 IF S$ = "AO" THEN 2360
2350 AMT = AMT + TIM
2360 FV = AMT*(1+JA)
2370 FX = FV
2380 FV = (INT((FV+5E-03)*100)/100)
2390 IF S$ = "AD" THEN 2430
2400 FV = FV + TIM
2410 AMT = FX + TIM
2420 GOTO 2440
2430 AMT = FX
2440 IF IPR$ <> "S" THEN 2470
2450 LPRINT "ANO ";I;SPC(10-LEN(STR$(I)));FV
2460 GOTO 2480
2470 PRINT "ANO ";I;SPC(10-LEN(STR$(I)));FV
2480 NEXT I
2490 PRINT
2500 INPUT "TECLE RETURN PARA CONTINUAR ";X$
2510 GOTO 140
2520 CLS:END
```

```
INVESTIMENTO COMPOSTO MENSAL
TOTAL DO INVESTIMENTO:  10000
TAXA DE JURO ANUAL:  23
NO. DE MESES:  12

MES  1           10191.67
MES  2           10387.01
MES  3           10586.09
MES  4           10788.99
MES  5           10995.78
MES  6           11206.53
MES  7           11421.32
MES  8           11640.23
MES  9           11863.33
MES  10          12090.71
MES  11          12322.45
MES  12          12558.63
```

```
INVESTIMENTO COMPOSTO ANUAL
TOTAL DO INVESTIMENTO:  25000
TAXA DE JURO ANUAL:  .2
NO. DE ANOS:  24
VALOR APOS  24  ANOS 1987421.18

ANO  1           30000
ANO  2           36000
ANO  3           43200
ANO  4           51840
ANO  5           62208
ANO  6           74649.6
ANO  7           89579.52
ANO  8           107495.42
ANO  9           128994.5
ANO  10          154793.4
ANO  11          185752.08
ANO  12          222902.5
ANO  13          267483
ANO  14          320979.6
ANO  15          385175.52
ANO  16          462210.62
ANO  17          554652.74
ANO  18          665583.29
ANO  19          798699.95
ANO  20          958439.94
ANO  21          1150127.93
ANO  22          1380153.52
ANO  23          1656184.22
ANO  24          1987421.06
```

```
INVESTIMENTO MENSAL (ORDINARIO)
TOTAL DO INVESTIMENTO MENSAL:  35000
TAXA DE JURO ANUAL:  21
TOTAL ANUAL:  594455.73

MES  1           35000
MES  2           70612.5
MES  3           106848.22
MES  4           143718.06
MES  5           181233.13
MES  6           219404.71
MES  7           258244.29
MES  8           297763.57
MES  9           337974.43
MES  10          378888.98
MES  11          420519.54
MES  12          462878.63
MES  13          505979.01
MES  14          549833.64
MES  15          594455.73
```

```
INVESTIMENTO ANUAL (ORDINARIO)
TOTAL DO INVESTIMENTO ANUAL:  45000
TAXA DE JURO ANUAL:  .19
TOTAL DA ANUIDADE:  1111898.79

ANO  1           45000
ANO  2           98550
ANO  3           162274.5
ANO  4           238106.66
ANO  5           328346.92
ANO  6           435732.83
ANO  7           563522.07
ANO  8           715591.27
ANO  9           896553.61
ANO  10          1111898.79
```

Figura 8. Programa Investimentos: exemplos de saída do programa.


```
MENSALIDADE DEVIDA
TOTAL DO INVESTIMENTO MENSAL:  50000
TAXA DE JURO ANUAL:  21
TOTAL ANUAL:  2521676.84

MES  1           50875
MES  2           102640.31
MES  3           155311.52
MES  4           208904.47
MES  5           263435.3
MES  6           318920.42
MES  7           375376.52
MES  8           432820.61
MES  9           491269.97
MES  10          550742.2
MES  11          611255.19
MES  12          672827.15
MES  13          735476.63
MES  14          799222.47
MES  15          864083.86
MES  16          930080.33
MES  17          997231.73
MES  18          1065558.29
MES  19          1135080.56
MES  20          1205819.47
MES  21          1277796.31
MES  22          1351032.75
MES  23          1425550.82
MES  24          1501372.96
MES  25          1578521.98
MES  26          1657021.12
MES  27          1736893.99
MES  28          1818164.63
MES  29          1900857.51
MES  30          1984997.52
MES  31          2070609.98
MES  32          2157720.65
MES  33          2246355.76
MES  34          2336541.99
MES  35          2428306.47
MES  36          2521676.84
```

```
ANUIDADE DEVIDA
TOTAL DO INVESTIMENTO ANUAL:  65000
TAXA DE JURO ANUAL:  .22
TOTAL DA ANUIDADE:  18872549.37

ANO  1           79300
ANO  2           176046
ANO  3           294076.12
ANO  4           438072.87
ANO  5           613748.9
ANO  6           828073.65
ANO  7           1089549.86
ANO  8           1408550.83
ANO  9           1797732.01
ANO  10          2272533.05
ANO  11          2851790.32
ANO  12          3558484.19
ANO  13          4420650.72
ANO  14          5472493.87
ANO  15          6755742.53
ANO  16          8321305.88
ANO  17          10231293.17
ANO  18          12561477.67
ANO  19          15404302.76
ANO  20          18872549.37
```

Figura 9. Programa Investimentos: exemplos de saída do programa (cont.).

MANUTENÇÃO DE DADOS

O programa **Manutenção de dados (TOTAIS)** pode realizar uma variedade de manutenções de escores. Ele foi desenvolvido para manter o registro de muitos totais diferentes. As categorias utilizadas pelo programa são definidas no comando DATA nas linhas 250 - 330. Você não está limitado a este número de ítens; até 50 classes com as quais os escores são acumulados. Pode-se ter várias cópias deste programa com diferentes dados. Vários totais podem ser salvos em fita ou disco e mais tarde adicionados regularmente ou não.

As cinco matrizes são usadas para acumular dados:

1. O número (deve estar entre 01 e 50) no grupo *(n)*.
2. O nome da categoria ou grupo (N$).
3. O último escore ou total da categoria (SL).
4. O novo escore a ser adicionado a SL (SN).
5. O novo total ou acumulado de SL e SN (NWTTL).

O comando DIM nas linhas 130 - 170 cria estas matrizes e o *loop* nas linhas 180 - 240 cria os valores numéricos com zero e os nomes com " @ " para todos. A linha 340 coloca o contador em 0. Esta variável mantém o registro do número de ítens colocados, de modo que a classificação e impressão seguinte saibam quantos ítens estão disponíveis. Lembre-se de que até 50 são possíveis; se você tiver mais do que 50 ítens, pode usar programas com diferentes comandos **DATA** para inserir dados em dois ou mais grupos. Nas linhas 360 - 480, você digita a informação necessária. Você deve digitar um escore anterior para cada (mesmo que ambos sejam zero). O programa perguntará pelo velho e o novo total para cada número de 1 para o **CTR** (o número total de comandos **DATA** numerados de 1 a 50, ou com números tão grandes quanto você queira).

A linha 500 pergunta pelo nome do seu relatório (não use vírgulas, a menos que você coloque o título entre aspas). A linha 510 pergunta se você quer imprimir. (Se não quiser, o relatório será mostrado na tela).

Três ordens são impressas: seqüência numérica por número (021 até 50), ordem descendente por escore total e ordem descendente por escore atual somente. Cada relatório é precedido por uma classificação do tipo bolha que organiza os dados na seqüência própria pela troca e união de dados que estão fora de ordem na matriz. Cada relatório é gerado pela mesma subrotina **GOSUB** 940 que imprime um subtítulo diferente baseado no valor de T2$ (mostrado nas linhas 520, 660 e 740).

Se o relatório for mostrado na tela, o movimento da tela será interrompido para manter o registro de quantos totais foram impressos (C2 nas linhas 1.170 - 1.190). Desta maneira, quando uma tela for preenchida, o programa esperará que você

pressione RETURN antes dele voltar a ativar a tela. Você pode repetir o relatório novamente e assim por diante até responder S para o comando de entrada da linha 760.

Note que nas rotinas de classificação, a mesma subrotina, GOSUB 960, é usada para trocar elementos que estão fora de ordem.

```
10 REM ***********************
20 REM *                     *
30 REM * MANUTENCAO DE DADOS *
40 REM *                     *
50 REM *  DIREITOS AUTORAIS  *
60 REM *                     *
70 REM *    NELSON ALVES     *
80 REM *                     *
90 REM *                     *
100 REM ***********************
110 KEY OFF
130 DIM N(50)
140 DIM N$(50)
150 DIM SL(50)
160 DIM SN(50)
170 DIM NTL(50)
180 FOR I = 1 TO 50
190 N$(I) = "@"
200 SL(I) = 0
210 SN(I) = 50
220 N(I) = I
230 NTL(I) = 0
240 NEXT I
250 DATA "01 NELSON & LUCIANE ALVES"
260 DATA "02 WILSON & DEBORAH FERREIRA"
270 DATA "03 RICARDO & TERESA ANDRADE"
280 DATA "04 ELIO & CLAUDINA FERREIRA"
290 DATA "05 TONY & MARGARETH ASSUMPCAO"
300 DATA "06 AGOSTINHO & JANETE ALVES"
310 DATA "07 FABIO & DEBORA PINI"
320 DATA "08 JOSE & CARMIM"
330 DATA "FIM"
340 CTR = 0
350 CLS
360 FOR I = 1 TO 50
370 READ NOME$
```

```
380 IF NOME$ = "FIM" THEN 420
390 CTR = CTR + 1
400 N$(I) = NOME$
410 NEXT I
420 FOR I = 1 TO CTR
430 PRINT "ENTRE O NO.: ";I
440 PRINT N$(I)
450 INPUT "ULTIMO TOTAL: ";SL(I)
460 INPUT "NOVO ESCORE: ";SN(I)
470 NTL(I) = SL(I) + SN(I)
480 NEXT I
490 REM *************************************
500 INPUT "TITULO DO RELATORIO: ";T$
510 INPUT "IMPRIMIR (S/N) ";ISN$
520 T2$ = "SEQUENCIA NUMERICA"
530 FLAG$ = "N"
540 FOR I = 1 TO CTR-1
550 IF N(I) <= N(I+1) THEN 570
560 GOSUB 790
570 NEXT I
580 IF FLAG$ = "S" THEN 530
590 GOSUB 960
600 FLAG$ = "N"
610 FOR I = 1 TO CTR-1
620 IF NTL(I) >= NTL(I+1) THEN 640
630 GOSUB 790
640 NEXT I
650 IF FLAG$ = "S" THEN 600
660 T2$ = "NOVA SEQUENCIA"
670 GOSUB 960
680 FLAG$ = "N"
690 FOR I = 1 TO CTRL-1
700 IF SN(I) >= SN(I+1) THEN 720
710 GOSUB 790
720 NEXT I
730 IF FLAG$ = "S" THEN 680
740 T2$ = "ESCORE ATUAL"
750 GOSUB 960
760 INPUT "REPETIR (S/N) ";RSN$
770 IF RSN$ = "S" THEN 520
780 GOTO 1340
790 NXTTL = NTL(I)
800 NTL(I) = NTL(I+1)
810 NTL(I+1) = NXTTL
820 NZ = N(I)
```



```
830 N(I) = N(I+1)
840 N(I+1) = NZ
850 NX$ = N$(I)
860 N$(I) = N$(I+1)
870 N$(I+1) = NX$
880 SXL = SL(I)
890 SL(I) = SL(I+1)
900 SL(I+1) = SXL
910 XSN = SN(I)
920 SN(I) = SN(I+1)
930 SN(I+1) = XSN
940 FLAG$ = "S"
950 RETURN
960 INPUT "PRESSIONE RETURN PARA CONTINUAR ";X$
970 C2 = 0
980 CLS
990 IF ISN$ <> "S" THEN 1030
1000 LPRINT T$
1010 LPRINT T2$
1020 LPRINT
1030 PRINT T$
1040 PRINT T2$
1050 PRINT
1060 IF ISN$ <> "S" THEN 1160
1070 LPRINT "****************************************";
1080 LPRINT "****************************************"
1090 LPRINT
1100 LPRINT T$
1110 LPRINT T2$
1120 LPRINT
1130 LPRINT SPC(40);"ANTERIOR";SPC(5);"NOVO";
1140 LPRINT SPC(5);"ATUAL"
1150 LPRINT
1160 FOR I = 1 TO CTR
1170 C2 = C2 + 1
1180 IF C2 < 5 THEN 1210
1190 C2 = 1
1200 INPUT "PRESSIONE RETURN PARA CONTINUAR ";X$
1210 PRINT I;SPC(2);N$(I)
1220 PRINT "TOTAL ANTERIOR: ";SL(I)
1230 PRINT "NOVO TOTAL: ";NTL(I)
1240 PRINT "TOTAL ATUAL: ";SN(I)
1250 PRINT "****************************************"
1260 IF ISN$ <> "S" THEN 1300
1270 LPRINT I;" - ";N$(I);SPC(36 - LEN(N$(I)));SL(I);
```

```
1280 LPRINT SPC(10-LEN(STR$(SL(I))));NTL(I);
1290 LPRINT SPC(9 - LEN(STR$(NTL(I))));SN(I)
1300 NEXT I
1310 IF ISN$ <> "S" THEN 1330
1320 LPRINT
1330 RETURN
1340 CLS: END
```



```
TESTE DO PROGRAMA
SEQUENCIA NUMERICA

**************************************************************************

TESTE DO PROGRAMA
SEQUENCIA NUMERICA
                                          ANTERIOR     NOVO      ATUAL

 1  - 01 NELSON & LUCIANE ALVES              1           3         2
 2  - 02 WILSON & DEBORAH FERREIRA           3           7         4
 3  - 03 RICARDO & TERESA ANDRADE            5          11         6
 4  - 04 ELIO & CLAUDINA FERREIRA            7          15         8
 5  - 05 TONY & MARGARETH ASSUMPCAO          9          10         1
 6  - 06 AGOSTINHO & JANETE ALVES            2           5         3
 7  - 07 FABIO & DEBORA PINI                 4           9         5
 8  - 08 JOSE & CARMIM                       6          13         7

TESTE DO PROGRAMA
NOVA SEQUENCIA

**************************************************************************

TESTE DO PROGRAMA
NOVA SEQUENCIA
                                          ANTERIOR     NOVO      ATUAL

 1  - 04 ELIO & CLAUDINA FERREIRA            7          15         8
 2  - 08 JOSE & CARMIM                       6          13         7
 3  - 03 RICARDO & TERESA ANDRADE            5          11         6
 4  - 05 TONY & MARGARETH ASSUMPCAO          9          10         1
 5  - 07 FABIO & DEBORA PINI                 4           9         5
 6  - 02 WILSON & DEBORAH FERREIRA           3           7         4
 7  - 06 AGOSTINHO & JANETE ALVES            2           5         3
 8  - 01 NELSON & LUCIANE ALVES              1           3         2

TESTE DO PROGRAMA
ESCORE ATUAL

**************************************************************************

TESTE DO PROGRAMA
ESCORE ATUAL
                                          ANTERIOR     NOVO      ATUAL

 1  - 04 ELIO & CLAUDINA FERREIRA            7          15         8
 2  - 08 JOSE & CARMIM                       6          13         7
 3  - 03 RICARDO & TERESA ANDRADE            5          11         6
 4  - 07 FABIO & DEBORA PINI                 4           9         5
 5  - 02 WILSON & DEBORAH FERREIRA           3           7         4
 6  - 06 AGOSTINHO & JANETE ALVES            2           5         3
 7  - 01 NELSON & LUCIANE ALVES              1           3         2
 8  - 05 TONY & MARGARETH ASSUMPCAO          9          10         1
```

Figura 10. Programa Manutenção de Dados: exemplo de saída do programa.

INFORMAÇÕES ÚTEIS

Cassete Com os Programas do Livro

Para obter a fita cassete opcional da EBRAS/STAFF contendo os 22 programas deste livro — prontos para utilização em seu micro — telefone para (011) 549.8633 ou escreva para EBRAS Editora Brasileira, Rua Borges Lagoa, 1.044 — Cep 04038 — Vila Mariana — São Paulo SP.

Preço: Cz$ 150,00

Software Personalizado, Processamento e Consultoria Especializada

Staff Consultoria e Processamento Ltda.
Rua Pascal, 799 — Cep 04616 — São Paulo - SP.
Telefone: (011) 530-3741

Impressão e acabamento
(com filmes fornecidos):
EDITORA SANTUÁRIO
Fone (0125) 36-2140
APARECIDA - SP

MELHORES PROGRAMAS

MSX — Melhores Programas (incluindo BASIC-MSX avançado) apresenta uma coletânea dos 22 "melhores programas" para aplicação imediata no computador MSX e compatível.

Dividido em 3 áreas de interesse, este livro sugere uma abordagem prática e muitas vezes avançada do BASIC-MSX em Jogos, Resolução de Problemas Matemáticos, Programas Práticos, Pessoais e Educacionais.

NELSON ALVES é Escritor Profissional, tendo traduzido e adaptado diversas obras para a EBRAS, autor do CP/M Incluindo Introdução aos Micros, CP/M-86 e MP/M (EBRAS), Engenheiro Eletrônico, Professor da Cadeira de Computadores Analógicos e Digitais da Faculdade de Tecnologia da Universidade Mackenzie-SP, Diretor da Staff Consultoria e Processamento Ltda.